# HP-IL PRINTER

```
VASM ROM ASSEMBLY            REV.  6/81A

OPTIONS: L C S

      2                       FILE   SCPR1B
      3     0              35 CON    @00035               ROM ID 0029
      4     1              33 CON    @00033               FUNCS+LABEL0026
****************************************************************************
* THE SWITCH ON THE PIL MODULE CAN CHANGE THIS ROM ADDRESS TO @40000*
* THE FIRST 4 WORDS OF THIS ROM HAS BEEN ARRANGED TO HANDLE THIS     *
* CASE, SO DON'T CHANGE THE NUMBER OF FUNCTIONS AND DON'T MOVE THE   *
* THE ACTUAL LOCATION OF THE HEADER UNLESS YOU KNOW WHAT YOU ARE     *
* DOING.                                              STEVE CHOU     *
****************************************************************************
     12     2               0 DEFP4K PHEAD              0
     12     3             400
     13     4               0 DEFR4K ACA                1
     13     5               0
     14     6               0 DEFR4K ACCHR              2
     14     7               0
     15    10               0 DEFR4K ACCOL              3
     15    11               0
     16    12               0 DEFR4K ACSPEC             4
     16    13               0
     17    14               0 DEFR4K ACX                5
     17    15               0
     18    16               0 DEFR4K BLDSPC             6
     18    17               0
     19    20               0 DEFR4K LIST               7
     19    21               0
     20    22               0 DEFR4K PRA                8
     20    23               0
     21    24            1000 U4KDEF PRAXIS             9
     21    25               0
     22    26               0 DEFR4K PRBUF             10
     22    27               0
     23    30               0 DEFR4K PRFLAG            11
     23    31               0
     24    32               0 DEFR4K PRKEYS            12
     24    33               0
     25    34               0 DEFR4K PRP               13
     25    35               0
     26    36            1000 U4KDEF PRPLOT            14
     26    37               0
     27    40            1000 U4KDEF PRPLTP            15
     27    41               0
     28    42               0 DEFR4K PRREG             16
     28    43               0
     29    44               0 DEFR4K PRREGX            17
     29    45               0
     30    46               0 DEFR4K PRSIGM            18
     30    47               0
     31    50               0 DEFR4K PRSTK             19
     31    51               0
     32    52               0 DEFR4K PRX               20
     32    53               0
     33    54               0 DEFR4K REGPLT            21
     33    55               0
     34    56               0 DEFR4K SKPCHR            22
```

```
  34    57               0
  35    60             · 0 DEFR4K SKPCOL                 23
  35    61               0
  36    62               0 DEFR4K STKPLT                 24
  36    63               0
  37    64               0 DEFR4K FMT                    25
  37    65               0
  38    66               0 DEFR4K PRNOP
  38    67               0
  39    70               0 CON     @00000
  40    71               0 CON     @00000
*
  42    72            1740 RTN                              FOR ROM ADDR SWITCH TO @40000
*
  44                       ENTRY   PRPLOT
  45                       ENTRY   PRPLTP
  46                       ENTRY   PRAXIS
  47    73             115 CON     @00115                 REGISTERS: 0077
  48    74            1140 CON     @01140                 BYTES 1ST REG 006
  49    75 PRPLOT      710 CON     @00710              0001 LBL  PRPLOT
  50    76               0 CON     @00000
  51    77             367 CON     @00367
  52   100               0 CON     @00000
  53   101             120 CON     @00120
  54   102             122 CON     @00122
  55   103             120 CON     @00120
  56   104             114 CON     @00114
  57   105             117 CON     @00117
  58   106             124 CON     @00124
  59   107             614 CON     @00614              0002 AON
  60   110             766 CON     @00766              0003 @NAME ?
  61   111             116 CON     @00116
  62   112             101 CON     @00101
  63   113             115 CON     @00115
  64   114             105 CON     @00105
  65   115              40 CON     @00040
  66   116              77 CON     @00077
  67   117             616 CON     @00616              0004 PRMT
  68   120             613 CON     @00613              0005 AOFF
  69   121             632 CON     @00632              0006 ASTO  11
  70   122              13 CON     @00013
  71   123             414 CON     @00414              0007 LBL  11
  72   124             767 CON     @00767              0008 @Y MIN ?
  73   125             131 CON     @00131
  74   126              40 CON     @00040
  75   127             115 CON     @00115
  76   130             111 CON     @00111
  77   131             116 CON     @00116
  78   132              40 CON     @00040
  79   133              77 CON     @00077
  80   134             616 CON     @00616              0009 PRMT
  81   135             460 CON     @00460              0010 STO 00
  82   136             767 CON     @00767              0011 @Y MAX ?
  83   137             131 CON     @00131
  84   140              40 CON     @00040
  85   141             115 CON     @00115
  86   142             101 CON     @00101
  87   143             130 CON     @00130
  88   144              40 CON     @00040
  89   145              77 CON     @00077
```

```
 90  146        616 CON     @00616
 91  147        461 CON     @00461
 92  150        506 CON     @00506
 93  151        674 CON     @00674
 94  152         30 CON     @00030
 95  153        415 CON     @00415
 96  154        766 CON     @00766
 97  155        101 CON     @00101
 98  156        130 CON     @00130
 99  157        111 CON     @00111
100  160        123 CON     @00123
101  161         40 CON     @00040
102  162         77 CON     @00077
103  163        651 CON     @00651
104  164         27 CON     @00027
105  165        616 CON     @00616
106  166        464 CON     @00464
107  167        654 CON     @00654
108  170         27 CON     @00027
109  171        632 CON     @00632
110  172          4 CON     @00004
111  173        441 CON     @00441
112  174        504 CON     @00504
113  175        675 CON     @00675
114  176         24 CON     @00024
115  177        567 CON     @00567
116  200        440 CON     @00440
117  201        505 CON     @00505
118  202        675 CON     @00675
119  203         31 CON     @00031
120  204        416 CON     @00416
121  205        767 CON     @00767
122  206        130 CON     @00130
123  207         40 CON     @00040
124  210        115 CON     @00115
125  211        111 CON     @00111
126  212        116 CON     @00116
127  213         40 CON     @00040
128  214         77 CON     @00077
129  215        616 CON     @00616
130  216        470 CON     @00470
131  217        767 CON     @00767
132  220        130 CON     @00130
133  221         40 CON     @00040
134  222        115 CON     @00115
135  223        101 CON     @00101
136  224        130 CON     @00130
137  225         40 CON     @00040
138  226         77 CON     @00077
139  227        616 CON     @00616
140  230        471 CON     @00471
141  231        506 CON     @00506
142  232        676 CON     @00676
143  233         30 CON     @00030
144  234        767 CON     @00767
145  235        130 CON     @00130
146  236         40 CON     @00040
147  237        111 CON     @00111
148  240        116 CON     @00116
149  241        103 CON     @00103
```

```
0012 PRNT
0013 STO 01
0014 X<=Y
0015 GTO 11

0016 LBL 12
0017 @AXIS ?

0018 CF    23

0019 PRNT
0020 STO 04
0021 FS?   23

0022 ASTO  04

0023 RCL 01
0024 X<Y?
0025 GTO 12

0026 CLX
0027 RCL 00
0028 X>Y?
0029 GTO 12

0030 LBL 13
0031 @X MIN ?

0032 PRNT
0033 STO 08
0034 @X MAX ?

0035 PRNT
0036 STO 09
0037 X<=Y
0038 GTO 13

0039 @X INC ?
```

```
150  242            40 CON     @00040
151  243            77 CON     @00077
152  244           616 CON     @00616          0040 PRMT
153  245           472 CON     @00472          0041 STO 10
154  246 PRPLTP    700 CON     @00700          0042 LBL   PRPLOTP
155  247            17 CON     @00017
156  250           370 CON     @00370
157  251             0 CON     @00000
158  252           120 CON     @00120
159  253           122 CON     @00122
160  254           120 CON     @00120
161  255           114 CON     @00114
162  256           117 CON     @00117
163  257           124 CON     @00124
164  260           120 CON     @00120
165  261           651 CON     @00651          0043 CF     12
166  262            14 CON     @00014
167  263           617 CON     @00617          0044 ADVN
168  264           426 CON     @00426          0045 6
169  265           647 CON     @00647          0046 XROM     2922
170  266           126 CON     @00126
171  267           770 CON     @00770          0047 @PLOT OF
172  270           120 CON     @00120
173  271           114 CON     @00114
174  272           117 CON     @00117
175  273           124 CON     @00124
176  274            40 CON     @00040
177  275           117 CON     @00117
178  276           106 CON     @00106
179  277            40 CON     @00040
180  300           633 CON     @00633          0048 ARCL  11
181  301            13 CON     @00013
182  302           647 CON     @00647          0049 XROM     2901
183  303           101 CON     @00101
184  304           647 CON     @00647          0050 XROM     2910
185  305           112 CON     @00112
186  306           450 CON     @00450          0051 RCL  08
187  307           451 CON     @00451          0052 RCL  09
188  310           761 CON     @00761          0053 @X
189  311           130 CON     @00130
190  312           741 CON     @00741          0054 XEQ    09
191  313            64 CON     @00064
192  314           211 CON     @00211
193  315           467 CON     @00467          0055 STO  07
194  316           427 CON     @00427          0056 7
195  317           647 CON     @00647          0057 XROM     2902
196  320           102 CON     @00102
197  321           647 CON     @00647          0058 XROM     2910
198  322           112 CON     @00112
199  323           421 CON     @00421          0059 1
200  324            23 CON     @00023               3
201  325            20 CON     @00020               0
202  326           462 CON     @00462          0060 STO  02
203  327           647 CON     @00647          0061 XROM     2909
204  330           111 CON     @00111
205  331           452 CON     @00452          0062 RCL  10
206  332           544 CON     @00544          0063 X>0?
207  333           661 CON     @00661          0064 GTO  00
208  334           207 CON     @00207
209  335           451 CON     @00451          0065 RCL  09
```

```
210  336           450 CON     @00450
211  337           501 CON     @00501
212  340           452 CON     @00452
213  341           541 CON     @00541
214  342           503 CON     @00503
215  343           472 CON     @00472
216  344           401 CON     @00401
217  345           451 CON     @00451
218  346           450 CON     @00450
219  347           541 CON     @00541
220  350           504 CON     @00504
221  351           561 CON     @00561
222  352           447 CON     @00447
223  353           503 CON     @00503
224  354           526 CON     @00526
225  355           550 CON     @00550
226  356           422 CON     @00422
227  357           501 CON     @00501
228  360           465 CON     @00465
229  361           450 CON     @00450
230  362           466 CON     @00466
231  363           417 CON     @00417
232  364           634 CON     @00634
233  365           205 CON     @00205
234  366           447 CON     @00447
235  367           503 CON     @00503
236  370           556 CON     @00556
237  371           647 CON     @00647
238  372           105 CON     @00105
239  373           423 CON     @00423
240  374           647 CON     @00647
241  375           127 CON     @00127
242  376           446 CON     @00446
243  377           656 CON     @00656
244  400           213 CON     @00213
245  401           647 CON     @00647
246  402           125 CON     @00125
247  403           452 CON     @00452
248  404           622 CON     @00622
249  405             6 CON     @00006
250  406           451 CON     @00451
251  407           446 CON     @00446
252  410           506 CON     @00506
253  411           677 CON     @00677
254  412            30 CON     @00030
255  413           634 CON     @00634
256  414             4 CON     @00004
257  415           605 CON     @00605
258  416 PRAXIS    714 CON     @00714
259  417            16 CON     @00016
260  420           367 CON     @00367
261  421             0 CON     @00000
262  422           120 CON     @00120
263  423           122 CON     @00122
264  424           101 CON     @00101
265  425           130 CON     @00130
266  426           111 CON     @00111
267  427           123 CON     @00123
268  430           651 CON     @00651
269  431            14 CON     @00014
```

```
0066 RCL 08
0067 -
0068 RCL 10
0069 ABS
0070 /
0071 STO 10
0072 LBL 00
0073 RCL 09
0074 RCL 08
0075 ABS
0076 X<Y?
0077 X<>Y
0078 RCL 07
0079 /
0080 LOG
0081 INT
0082 2
0083 -
0084 STO 05
0085 RCL 08
0086 STO 06
0087 LBL 14
0088 FIX I 05

0089 RCL 07
0090 /
0091 RND
0092 XROM    2905

0093 3
0094 XROM    2923

0095 RCL 06
0096 XEQ I 11

0097 XROM    2921

0098 RCL 10
0099 STO+  06

0100 RCL 09
0101 RCL 06
0102 X<=Y
0103 GTO 14

0104 FIX   04

0105 RTN
0106 LBL  PRAXIS




0107 CF    12
```

```
270  432        440 CON    @00440        0108 RCL  00
271  433        441 CON    @00441        0109 RCL  01
272  434        761 CON    @00761        0110 @Y
273  435        131 CON    @00131
274  436        740 CON    @00740        0111 XEQ    09
275  437        340 CON    @00340
276  440        211 CON    @00211
277  441        466 CON    @00466        0112 STO  06
278  442        421 CON    @00421        0113 1
279  443         22 CON    @00022             2
280  444         25 CON    @00025             5
281  445        647 CON    @00647        0114 XROM     2902
282  446        102 CON    @00102
283  447        647 CON    @00647        0115 XROM     2910
284  450        112 CON    @00112
285  451        442 CON    @00442        0116 RCL  02
286  452        550 CON    @00550        0117 INT
287  453        541 CON    @00541        0118 ABS
288  454        462 CON    @00462        0119 STO  02
289  455        421 CON    @00421        0120 1
290  456         26 CON    @00026             6
291  457         30 CON    @00030             8
292  460        504 CON    @00504        0121 X<Y?
293  461        673 CON    @00673        0122 GTO  10(UNCOMPILED)
294  462          0 CON    @00000
295  463        440 CON    @00440        0123 RCL  00
296  464        446 CON    @00446        0124 RCL  06
297  465        503 CON    @00503        0125 /
298  466        556 CON    @00556        0126 RND
299  467        647 CON    @00647        0127 XROM     2905
300  470        105 CON    @00105
301  471        740 CON    @00740        0128 XEQ    05
302  472        220 CON    @00220
303  473        205 CON    @00205
304  474        564 CON    @00564        0129 R^
305  475        441 CON    @00441        0130 RCL  01
306  476        740 CON    @00740        0131 XEQ    04
307  477        207 CON    @00207
308  500        204 CON    @00204
309  501        564 CON    @00564        0132 R^
310  502        500 CON    @00500        0133 +
311  503        501 CON    @00501        0134 -
312  504        427 CON    @00427        0135 7
313  505        506 CON    @00506        0136 X<=Y
314  506        565 CON    @00565        0137 RDWN
315  507        647 CON    @00647        0138 XROM     2923
316  510        127 CON    @00127
317  511        441 CON    @00441        0139 RCL  01
318  512        446 CON    @00446        0140 RCL  06
319  513        503 CON    @00503        0141 /
320  514        556 CON    @00556        0142 RND
321  515        647 CON    @00647        0143 XROM     2905
322  516        105 CON    @00105
323  517        617 CON    @00617        0144 ADVN
324  520        444 CON    @00444        0145 RCL  04
325  521        572 CON    @00572        0146 SIGN
326  522        547 CON    @00547        0147 X=0?
327  523        664 CON    @00664        0148 GTO  03
328  524        317 CON    @00317
329  525        566 CON    @00566        0149 LSTX
```

```
330  526        440 CON     @00440        0150 RCL  00
331  527        505 CON     @00505        0151 X>Y?
332  530        673 CON     @00673        0152 GTO  10(UNCOMPILED)
333  531          0 CON     @00000
334  532        501 CON     @00501        0153 -
335  533        441 CON     @00441        0154 RCL  01
336  534        440 CON     @00440        0155 RCL  00
337  535        501 CON     @00501        0156 -
338  536        504 CON     @00504        0157 X<Y?
339  537        673 CON     @00673        0158 GTO  10(UNCOMPILED)
340  540          0 CON     @00000
341  541        503 CON     @00503        0159 /
342  542        442 CON     @00442        0160 RCL  02
343  543        421 CON     @00421        0161 1
344  544        501 CON     @00501        0162 -
345  545        502 CON     @00502        0163 *
346  546        432 CON     @00432        0164 .
347  547         25 CON     @00025             5
348  550        500 CON     @00500        0165 +
349  551        550 CON     @00550        0166 INT
350  552        621 CON     @00621        0167 STO    Y
351  553        162 CON     @00162
352  554        444 CON     @00444        0168 RCL  04
353  555        446 CON     @00446        0169 RCL  06
354  556        503 CON     @00503        0170 /
355  557        556 CON     @00556        0171 RND
356  560        647 CON     @00647        0172 XROM     2905
357  561        105 CON     @00105
358  562        740 CON     @00740        0173 XEQ    05
359  563        127 CON     @00127
360  564        205 CON     @00205
361  565        422 CON     @00422        0174 2
362  566        503 CON     @00503        0175 /
363  567        505 CON     @00505        0176 X>Y?
364  570        661 CON     @00661        0177 GTO  00
365  571        211 CON     @00211
366  572        500 CON     @00500        0178 +
367  573        442 CON     @00442        0179 RCL  02
368  574        421 CON     @00421        0180 1
369  575        501 CON     @00501        0181 -
370  576        504 CON     @00504        0182 X<Y?
371  577        603 CON     @00603        0183 ENT^
372  600        501 CON     @00501        0184 -
373  601        662 CON     @00662        0185 GTO  01
374  602        205 CON     @00205
375  603        401 CON     @00401        0186 LBL  00
376  604        603 CON     @00603        0187 ENT^
377  605        500 CON     @00500        0188 +
378  606        442 CON     @00442        0189 RCL  02
379  607        501 CON     @00501        0190 -
380  610        402 CON     @00402        0191 LBL  01
381  611        647 CON     @00647        0192 XROM     2923
382  612        127 CON     @00127
383  613        617 CON     @00617        0193 ADVN
384  614        740 CON     @00740        0194 XEQ    08
385  615        152 CON     @00152
386  616        210 CON     @00210
387  617        465 CON     @00465        0195 STO  05
388  620        547 CON     @00547        0196 X=0?
389  621        661 CON     @00661        0197 GTO  00
```

```
390   622         225 CON     @00225
391   623         442 CON     @00442          0198 RCL  02
392   624         421 CON     @00421          0199 1
393   625         501 CON     @00501          0200 -
394   626         570 CON     @00570          0201 X=Y?
395   627         661 CON     @00661          0202 GTO  00
396   630         217 CON     @00217
397   631         561 CON     @00561          0203 X<>Y
398   632         421 CON     @00421          0204 1
399   633         501 CON     @00501          0205 -
400   634         740 CON     @00740          0206 XEQ    06
401   635          77 CON     @00077
402   636         206 CON     @00206
403   637         445 CON     @00445          0207 RCL  05
404   640         421 CON     @00421          0208 1
405   641         500 CON     @00500          0209 +
406   642         662 CON     @00662          0210 GTO  01
407   643         207 CON     @00207
408   644         404 CON     @00404          0211 LBL  03
409   645         740 CON     @00740          0212 XEQ    08
410   646         121 CON     @00121
411   647         210 CON     @00210
412   650         401 CON     @00401          0213 LBL  00
413   651         442 CON     @00442          0214 RCL  02
414   652         422 CON     @00422          0215 2
415   653         402 CON     @00402          0216 LBL  01
416   654         501 CON     @00501          0217 -
417   655         740 CON     @00740          0218 XEQ    06
418   656          56 CON     @00056
419   657         206 CON     @00206
420   660         617 CON     @00617          0219 ADVN
421   661         442 CON     @00442          0220 RCL  02
422   662         445 CON     @00445          0221 RCL  05
423   663         421 CON     @00421          0222 1
424   664         500 CON     @00500          0223 +
425   665         421 CON     @00421          0224 1
426   666          33 CON     @00033               EEX
427   667          23 CON     @00023               3
428   670         503 CON     @00503          0225 /
429   671         500 CON     @00500          0226 +
430   672         603 CON     @00603          0227 ENT^
431   673         524 CON     @00524          0228 CHS
432   674         561 CON     @00561          0229 X<>Y
433   675         444 CON     @00444          0230 RCL  04
434   676         572 CON     @00572          0231 SIGN
435   677         547 CON     @00547          0232 X=0?
436   700         565 CON     @00565          0233 RDWN
437   701         565 CON     @00565          0234 RDWN
438   702         462 CON     @00462          0235 STO  02
439   703         634 CON     @00634          0236 FIX    04
440   704           4 CON     @00004
441   705         605 CON     @00605          0237 RTN
442   706         405 CON     @00405          0238 LBL  04
443   707         446 CON     @00446          0239 RCL  06
444   710         503 CON     @00503          0240 /
445   711         556 CON     @00556          0241 RND
446   712         406 CON     @00406          0242 LBL  05
447   713         541 CON     @00541          0243 ABS
448   714         550 CON     @00550          0244 INT
449   715         543 CON     @00543          0245 X#0?
```

```
450  716        661 CON     @00661        0246 GTO 00
451  717        202 CON     @00202
452  720        565 CON     @00565        0247 RDWN
453  721        425 CON     @00425        0248 5
454  722        401 CON     @00401        0249 LBL 00
455  723        526 CON     @00526        0250 LOG
456  724        550 CON     @00550        0251 INT
457  725        445 CON     @00445        0252 RCL 05
458  726        500 CON     @00500        0253 +
459  727        423 CON     @00423        0254 3
460  730        500 CON     @00500        0255 +
461  731        427 CON     @00427        0256 7
462  732        502 CON     @00502        0257 *
463  733        605 CON     @00605        0258 RTN
464  734        407 CON     @00407        0259 LBL 06
465  735        603 CON     @00603        0260 ENT^
466  736        603 CON     @00603        0261 ENT^
467  737        427 CON     @00427        0262 7
468  740        513 CON     @00513        0263 MOD
469  741        422 CON     @00422        0264 2
470  742        503 CON     @00503        0265 /
471  743        550 CON     @00550        0266 INT
472  744        647 CON     @00647        0267 XROM    2923
473  745        127 CON     @00127
474  746        501 CON     @00501        0268 -
475  747        761 CON     @00761        0269 @-
476  750         55 CON     @00055
477  751        410 CON     @00410        0270 LBL 07
478  752        427 CON     @00427        0271 7
479  753        505 CON     @00505        0272 X>Y?
480  754        661 CON     @00661        0273 GTO 00
481  755        205 CON     @00205
482  756        501 CON     @00501        0274 -
483  757        647 CON     @00647        0275 XROM    2901
484  760        101 CON     @00101
485  761        670 CON     @00670        0276 GTO 07
486  762         12 CON     @00012
487  763        401 CON     @00401        0277 LBL 00
488  764        565 CON     @00565        0278 RDWN
489  765        647 CON     @00647        0279 XROM    2923
490  766        127 CON     @00127
491  767        411 CON     @00411        0280 LBL 08
492  770        421 CON     @00421        0281 1
493  771         22 CON     @00022             2
494  772         27 CON     @00027             7
495  773        647 CON     @00647        0282 XROM    2903
496  774        103 CON     @00103
497  775        564 CON     @00564        0283 R^
498  776        605 CON     @00605        0284 RTN
499  777        412 CON     @00412        0285 LBL 09
500 1000        771 CON     @00771        0286-@ <UNITS=
501 1001        177 CON     @00177
502 1002         40 CON     @00040
503 1003         74 CON     @00074
504 1004        125 CON     @00125
505 1005        116 CON     @00116
506 1006        111 CON     @00111
507 1007        124 CON     @00124
508 1010        123 CON     @00123
509 1011         75 CON     @00075
```

```
510 1012         506 CON    @00506          0287 X<=Y
511 1013         673 CON    @00673          0288 GTO 10
512 1014         303 CON    @00303
513 1015         561 CON    @00561          0289 X<>Y
514 1016         541 CON    @00541          0290 ABS
515 1017         504 CON    @00504          0291 X<Y?
516 1020         561 CON    @00561          0292 X<>Y
517 1021         526 CON    @00526          0293 LOG
518 1022         546 CON    @00546          0294 X<0?
519 1023         661 CON    @00661          0295 GTO 00
520 1024         213 CON    @00213
521 1025         550 CON    @00550          0296 INT
522 1026         422 CON    @00422          0297 2
523 1027         561 CON    @00561          0298 X<>Y
524 1030         505 CON    @00505          0299 X>Y?
525 1031         662 CON    @00662          0300 GTO 01
526 1032         215 CON    @00215
527 1033         501 CON    @00501          0301 -
528 1034         465 CON    @00465          0302 STO 05
529 1035         420 CON    @00420          0303 0
530 1036         663 CON    @00663          0304 GTO 02
531 1037         215 CON    @00215
532 1040         401 CON    @00401          0305 LBL 00
533 1041         551 CON    @00551          0306 FRAC
534 1042         543 CON    @00543          0307 X#0?
535 1043         421 CON    @00421          0308 1
536 1044         566 CON    @00566          0309 LSTX
537 1045         550 CON    @00550          0310 INT
538 1046         561 CON    @00561          0311 X<>Y
539 1047         501 CON    @00501          0312 -
540 1050         402 CON    @00402          0313 LBL 01
541 1051         763 CON    @00763          0314-@ E
542 1052         177 CON    @00177
543 1053          40 CON    @00040
544 1054         105 CON    @00105
545 1055         403 CON    @00403          0315 LBL 02
546 1056         424 CON    @00424          0316 4
547 1057         647 CON    @00647          0317 XROM     2922
548 1060         126 CON    @00126
549 1061         647 CON    @00647          0318 XROM     2901
550 1062         101 CON    @00101
551 1063         634 CON    @00634          0319 FIX    00
552 1064           0 CON    @00000
553 1065         565 CON    @00565          0320 RDWN
554 1066         547 CON    @00547          0321 X=0?
555 1067         661 CON    @00661          0322 GTO 00
556 1070         212 CON    @00212
557 1071         647 CON    @00647          0323 XROM     2905
558 1072         105 CON    @00105
559 1073         527 CON    @00527          0324 10^X
560 1074         422 CON    @00422          0325 2
561 1075         465 CON    @00465          0326 STO 05
562 1076         634 CON    @00634          0327 FIX    02
563 1077           2 CON    @00002
564 1100         565 CON    @00565          0328 RDWN
565 1101         662 CON    @00662          0329 GTO 01
566 1102         206 CON    @00206
567 1103         401 CON    @00401          0330 LBL 00
568 1104         421 CON    @00421          0331 1
569 1105         647 CON    @00647          0332 XROM     2905
```

```
   570 1106             105 CON     @00105
   571 1107             634 CON     @00634           0333 FIX I 05
   572 1110             205 CON     @00205
   573 1111             402 CON     @00402           0334 LBL 01
   574 1112             762 CON     @00762           0335 @>
   575 1113              76 CON     @00076
   576 1114              40 CON     @00040
   577 1115             647 CON     @00647           0336 XROM    2901
   578 1116             101 CON     @00101
   579 1117             605 CON     @00605           0337 RTN
   580 1120             413 CON     @00413           0338 LBL 10
   581 1121             420 CON     @00420           0339 0
   582 1122             503 CON     @00503           0340 /
   583 1123               0 CON     @00000                NULL*************
   584 1124             710 CON     @00710           0341 END
   585 1125              56 CON     @00056
   586 1126            1057 CON     @01057
*
*************************************************************
*SKPCHR-SKIP SPACES AS SPECIFIED BY X-23 MAX.
*************************************************************
   591                      ENTRY   SKPCHR
   592 1127             222 CON     @222
   593 1130              10 CON     @10
   594 1131               3 CON     3
   595 1132              20 CON     16
   596 1133              13 CON     11
   597 1134              23 CON     19
   598 1135 SKPCHR        1 GOSUB   CONV3D          GET X CONV TO BIN
   598 1136               0
   599 1137             406 A=C     X               SAVE BINARY X
   600 1140             460 LDI
   601 1141              30 CON     24
   602 1142            1406 ? A<C   X               # OF CHARS TO SKIP<24?
   603 1143             253 GONC    ERL     (1170)
   604 1144             216 B=A                     SAVE A IN B TEMP.
   605 1145               1 GOSUB   IACHR           INITIALIZE,SEND MODE IF NECESSAR
   605 1146               0
   606 1147             460 LDI
   607 1150             240 CON     @240
   608 1151             156 AB EX                   RESTORE A
   609 1152             210 S5=     1
   610 1153             513 GOTO    SKPC10 (1224)
***************************************************************************
******************   SKPCOL =  SKIP COLUMNS   *****************************
***************************************************************************
   614                      ENTRY   SKPCOL
   615 1154             214 CON     @214            L
   616 1155              17 CON     15              O
   617 1156               3 CON     3               C
   618 1157              20 CON     16              P
   619 1160              13 CON     11              K
   620 1161              23 CON     19              S
   621 1162 SKPCOL        1 GOSUB   CONV3D          GET ARGUMENT FROM XREG
   621 1163               0
   622 1164             406 A=C     X
   623 1165             460 LDI
   624 1166             250 CON     168
   625 1167            1406 ? A<C   X
   626 1170 ERL           1 GOLNC   ERRDE           #OF COLS TOO LARGE
```

```
1171                2
1172              216 B=A                          SAVE A IN B TEMP
1173                1 GOSUB  IACOL                 INITIALIZE,SEND MODE IF NECESSARY
1174                0
1175              156 AB EX                        RESTORE A
1176              210 S5=    1                     REMEMBER EXIT TO XPECHK
1177               23 GOTO   SKPC4  (1201)

*SKPCOM= SKIP COLUMN, MICROCODE
*USES:   A(X),C,N      NO STATUS,  NO PT,   1 ADDITIONAL SUB LEVEL
*INPUTS:  C(X)= # COLUMNS TO SKIP (SKPCOM)
*         A(X)= # COLUMNS TO SKIP (SKPC4)
*         PRINTER MODE ALREADY SET TO PROPER STATE
*IN&OUT:  HEX MODE

                        ENTRY  SKPCOM
                        ENTRY  SKPC4
1200 SKPCOM       406 A=C    X                     # COLS TO "A" (BINARY)
1201 SKPC4        460 LDI
1202              237 CON    @237                  (SKIP 0 CHAR) - 1
1203              674 RCR    11                    CHAR CTR TO C(M)
1204              460 LDI
1205                7 CON    7                     7 COLUMNS/CHARACTER
1206             1406 ? A<C  X                     # COLUMNS < 7?
1207              137 GOC    SKPC8  (1222) YES, DON'T SEND # CHAR
1210 SKPC6       1072 C=C+1  M                     ADD A CHARACTER
1211              706 A=A-C  X                     SUBTRACT 7 COLUMNS
1212             1763 GONC   SKPC6  (1210)
1213               74 RCR    3                     CHAR CTR TO C(X)
1214                1 GOSUB  PBYTEC                # BLANK CHARS TO PRINTER
1215                0
1216              674 RCR    11                    BRING BACK THE 7
1217              506 A=A+C  X                     RESTORE # COLUMNS
1220             1506 ? A#0  X                     # COLUMNS= 0?
1221               53 GONC   SKPC20 (1226) YES, DON'T SEND IT
1222 SKPC8        460 LDI
1223              270 CON    @270                  SKIP 0 COLUMNS
1224 SKPC10         1 GOSUB  PBYA+C                # BLANK COLUMNS TO PRINTER
1225                0
1226 SKPC20       214 ?S5=1                        EXIT TO XPECHK ?
1227             1640 RTN NC                       NO, RETURN TO CALLING PRGM
1230              143 GOTO   XPECHK (1244)
                      EJECT
```

```
**************************************************************************
**************** PRA -- PRINT ALPHA REG, NO DISPLAY  *****************
**************************************************************************
    669                       ENTRY  LPECHK
    670                       ENTRY  PRA
    671 1231           201 CON    @201          A
    672 1232            22 CON    @22           R
    673 1233            20 CON    @20           P
629C 674 1234 PRA        1 GOSUB  IPRT
    674 1235             0
 *
    676                       ENTRY  PRA20
 *
629E 678 1236 PRA20      1 GOSUB  PAREG
    678 1237             0
    679 1240          1670 C=REGN 14            RESTORE SS0 FOR AVIEW PATH
    680 1241          1530 ST=C
    681 1242 LPECHK      1 GOSUB  EOLL
    681 1243             0
62A4 682 1244 XPECHK     1 GOLONG PECHK
    682 1245             2
**************************************************************************
*********************** PRT 7= PROMPT ********************************
**************************************************************************
    686                       ENTRY  PPROMP
62A6 687 1246 PPROMP     1 GOSUB  CKEN
    687 1247             0
    688 1250          1740 RTN                  P+1 - DON'T PRINT
    689 1251           410 S8=    1             P+2
    690 1252             1 GOSUB  FNDPTR
    690 1253             0
    691 1254          1740 RTN                  PRINTER NOT FOUND
    692 1255             1 GOSUB  IAUNB
    692 1256             0
    693 1257          1740 RTN                  DON'T PRINT IN MANUAL MODE
    694 1260          1563 GOTO   PRA20  (1236) P+2 - PRINT
*************************************************************************
**************** ACA - ACCUMULATE ALPHA REGISTER *******************
*************************************************************************
    698                       ENTRY  ACA
    699 1261           201 CON    @201          A
    700 1262             3 CON    3             C
    701 1263             1 CON    1             A
    702 1264 ACA         1 GOSUB  IACHR  62B4
    702 1265             0
    703 1266             1 GOSUB  PAREG
    703 1267             0
    704 1270          1543 GOTO   XPECHK (1244)
************************************************************************
*-PAREG        SEND ALPHA REG TO PRINTER  62B9
*
*-USES:  A,B(X&S),C,N,  ACTIVE PT, S9 FOR ERRORS, 1 ADDITIONAL SUB LEVEL
*-INPUTS: CHIP 0 ENABLED,  HEXMODE
* OUTPUT: A.M=# OF CHARACTERS IN ALPHA REGISTER, PT=0 (CAN BE CHANGED),
*         CHIP 0 ENABLED,  HEX MODE
*
*
    714                       ENTRY  PAREG
    715 1271 PAREG     116 C=0
```

```
1272          1634 PT=      0
1273          1020 LC       8                C(X)= REG 8 ADDR
1274           220 LC       2                C(S)= REG BYTE CTR (R8= 3 BYTES)
1275           416 A=C                       A= 2 0000000000 008
1276          1334 PT=      13
1277           620 LC       6                C(S)= REG BYTE COUNTER
1300           376 CB EX    S                B.S = 6
1301          1070 C=REGN   8                GET REG 8
1302           574 RCR      6                1ST ALPHA REG BYTE TO C(0-1)
1303          1434 PT=      1
1304 PAR40    1574 RCR      12               NEXT BYTE TO C(0-1)
1305          1424 ? PT=    1                STILL LOOKING FOR 1ST CHAR?
1306            33 GONC     PAR60   (1311)   NO
1307          1352 ? C#0    WPT              YES, C(0-1)= 1ST CHAR?
1310            73 GONC     PAR70   (1317)   NO
1311 PAR60       1 GOSUB    CKANGL           CHECK IF THE CHAR IS A ANGEL SIGN
1312             0
1313             1 GOSUB    PBYTDU           SEND CHARACTER TO PRINTER
1314             0
1315          1634 PT=      0
1316           572 A=A+1    M                COUNT THE CHARACTER
1317 PAR70     676 A=A-1    S                DONE WITH REG YET?
1320          1643 GONC     PAR40   (1304)   NO
1321           176 A=B      S                YES, A(S)=6= REG BYTE CTR
1322           236
1323           646 A=A-1    X                GET NEXT REG ADDR
1324           246 C=A      X                COPY ADDR TO C
1325           406
1326          1160 DADD=C
1327           460 LDI
1330             5 CON      5
1331          1406 ? A<C    X                MORE REG TO CHECK ?
1332          1540 RTN C                     NO
1333            70 C=DATA
1334          1503 GOTO     PAR40   (1304)
                   EJECT
```

748 EJECT

```
*******************************************************************
************************************************************************
*PRSTK-PRINT STACK ROUTINE
*PRINTS STACK IN T,Z,Y,X ORDER.
*************************************************************************
   754                       ENTRY  PRSTK
   755                       ENTRY  PRSTKX
   756 1335            213 CON    @213
   757 1336             24 CON    @24
   758 1337             23 CON    @23
   759 1340             22 CON    @22
   760 1341             20 CON    @20
   761 1342 PRSTK        1 GOSUB  IPRT
   761 1343              0
   762 1344            660 C=STK                 GET RTN ADDR OF NFRPU
   763 1345           1172 C=C-1  M             CHANGE IT TO RTN TO NFRC
   764 1346            560 STK=C                 SET FOR NFRC
   765 1347 PRSTKX     116 C=0
   766 1350            460 LDI                   C.M=0,C.X=3
   767 1351              3 CON    @3
   768 1352           1150 REGN=C 9
   769 1353            773 GOTO   REGL00 (1452)
***********************************************************************
*PRREG-PRINT REGISTERS
***********************************************************************
   773                       ENTRY  PRREG
   774 1354            207 CON    @207
   775 1355              5 CON    @5
   776 1356             22 CON    @22
   777 1357             22 CON    @22
   778 1360             20 CON    @20
   779 1361 PRREG        1 GOSUB  FNDEND          FIND LAST REG
   779 1362              0
   780 1363            646 A=A-1  X
   781 1364            116 C=0
   782 1365           1160 DADD=C
   783 1366           1570 C=REGN 13              GET REG 0
   784 1367            272 AC EX  M
   785 1370            543 GOTO   REGL   (1444)
*************************************************************************
*PRSIGM-PRINT THE STATISTICS REGSITERS.
*************************************************************************
   789                       ENTRY  PRSIGM
   790 1371            316 CON    @316            SIGMA
   791 1372             22 CON    @22             R
   792 1373             20 CON    @20             P
   793 1374 PRSIGM       1 GOSUB  SUMCHK          STOP ADR IN C.X
   793 1375              0
   794 1376            246 AC EX  X               STOP ADR IN A.X
   795 1377            116 C=0                    ENABLE CHIP 0
   796 1400           1160 DADD=C                 (SUMCHK LEAVES IT DISABLED)
   797 1401           1570 C=REGN 13              GET SIGMA ADR
   798 1402            334 PT=    10              PUT IN A
   799 1403            112 C=0    WPT
   800 1404            474 RCR    8
   801 1405           1076 C=C+1  S               SIGMA FLAG SET
   802 1406            246 AC EX  X               START=C.M STOP=C.X
   803 1407 STKCKX     403 GOTO   STKCHK (1447) DO IT
*************************************************************************
```

```
*PPREGX-PRINT REGISTERS AS SPECIFIED BY THE X REGISTER.
**************************************************************************
  807                        ENTRY   PRREGX
  808 1410               230 CON     @230
  809 1411                 7 CON     @7
  810 1412                 5 CON     @5
  811 1413                22 CON     @22
  812 1414                22 CON     @22
  813 1415      .         20 CON     @20
  814 1416 PRREGX         1 GOSUB   CONV3D
  814 1417                 0
  815 1420               674 RCR     11
  816 1421              1150 REGN=C  9              STORE START ADDRESS
  817 1422              1240 SETDEC
  818 1423               370 C=REGN  3              GET X
  819 1424               204 S5=     0              SET FRACTION FLAG
  820 1425                 1 GOSUB   INTFRC         GET FRACTION OF X
  820 1426                 0
  821 1427              1046 C=C+1   X
  822 1430              1046 C=C+1   X
  823 1431              1046 C=C+1   X              MULT BY 1000
  824 1432              1140 SETHEX
  825 1433                 1 GOSUB   CONV3C         CONVERT FRAC TO BIN
  825 1434                 0
  826 1435               246 AC EX   X              PUT STOP NUMBER IN A
  827 1436              1170 C=REGN  9              START NUM IN C
  828 1437               272 AC EX   M              START ADR IN PLC
  829 1440              1570 C=REGN  13             GET REG 0
  830 1441               532 A=A+C   M              GEN ADR
  831 1442                74 RCR     3              MOVE REG 0
  832 1443               506 A=A+C   X
  833                        ENTRY   REGL
  834 1444 REGL          116 C=0                    CLEAR HGIH END
  835 1445               234 PT=     5
  836 1446               252 AC EX   WPT
  837 1447 STKCHK       1150 REGN=C  9              ENTRY FOR PRREG
  838 1450                 1 GOSUB   IPRT
  838 1451                 0
  839 1452 REGL00          1 GOSUB   EOLL           LINE FEED
  839 1453                 0
  840
  841                        ENTRY   REGLOP
  842 1454 REGLOP          1 GOSUB   UNL            SEND UNLISTEN
  842 1455                 0
  843 1456              1170 C=REGN  9              GET ADDRESS
  844 1457                74 RCR     3              ADR IN C.X
  845 1460                 1 GOSUB   CHKADR         ERRNE IF REG NONEXISTANT
  845 1461                 0
  846                                               C(X)= REG ADDR, B= REG CONTENTS
  847 1462              1104 S9=     0
  848 1463               356 BC EX                  GET VALUE BACK
  849 1464               530 M=C                    SAVE FOR LATER
  850 1465                 1 GOSUB   LISTEN         ADDRESS PRINTER AS A LISTENER
  850 1466                 0
  851 1467               116 C=0
  852 1470              1160 DADD=C
  853 1471              1170 C=REGN  9              GET N
  854 1472               256 AC EX
  855 1473              1570 C=REGN  13
  856 1474               234 PT=     5
```

```
1475              106 C=0      X
1476             1112 C=A-C    WPT              ADDRESS TO BYTE
1477              647 GOC      STK     (1563) IF CARRY THEN STACK ADR
1500             1536 ? A#0    S                IS THIS SIGMA REGISTERS?
1501              523 GONC     REG     (1553) NO
1502                1 GOSUB    SIGSTF           LOOK UP SIGMA ALPHA
1503                0
1504              176 CON      @176             SIGMA
1505              130 CON      @130             X
1506               40 CON      @40
1507             1040 CON      @1040
1510              176 CON      @176             SIGMA
1511              130 CON      @130             X^2
1512              136 CON      @136
1513             1062 CON      @1062
1514              176 CON      @176             SIGMA
1515              131 CON      @131             Y
1516               40 CON      @40
1517             1040 CON      @1040
1520              176 CON      @176             SIGMA
1521              131 CON      @131             Y^2
1522              136 CON      @136
1523             1062 CON      @1062
1524              176 CON      @176             SIGMA
1525              130 CON      @130             XY
1526              131 CON      @131
1527             1040 CON      @1040
1530              116 CON      @116             N
1531             1243 CON      @1243            THREE BLANKS
                      ENTRY    SIGSTF
1532 SIGSTF       106 C=0      X
1533              474 RCR      8                CALCULATE ADDR FOR TBLE
1534              732 A=A-C    M
1535              660 C=STK
1536             1032 C=C+A    M                ADD OFFSET 4 TIMES
1537             1032 C=C+A    M
1540             1032 C=C+A    M
1541             1032 C=C+A    M
1542 MORALP      1460 CXISA                     GET CHR
1543                1 GOSUB    CKANCL           CHECK IF TALKING TO T.V.
1544                0
1545                1 GOSUB    PBYTEC           PUT IT OUT
1546                0
1547             1072 C=C+1    M                INC COUNT
1550             1366 ?C#0     XS               LAST BYTE?
1551             1713 GONC     MORALP (1542) NO
1552              243 GOTO     MSG    (1576)
1553 REG          460 LDI                       LOAD R CONSTANT
1554              122 CON      @122             R
1555                1 GOSUB    PBYTEC
1556                0
1557               74 RCR      3                OUTPUT REG #
1560                1 GOSUB    PBINBO           REG # TO PRINTER
1561                0
1562              143 GOTO     MSG    (1576)
1563 STK            1 GOSUB    STKADR           TABLE CHARACTER LOOK UP
1564                0
1565              124 CON      @124             T
1566              132 CON      @132             Z
1567              131 CON      @131             Y
```

```
   911 1570           130 CON    @130         X
   912
   913                    ENTRY  STKADR
   914 1571 STKADR  660 C=STK               GET T,Z,Y,X
   915 1572          1032 C=C+A  M
   916 1573          1460 CXISA
   917 1574             1 GOSUB  PBYTEC
   917 1575             0
   918 1576 MSG         1 GOSUB  PRTMSG        "= " TO PRINTER
   918 1577             0
   919 1600            75 CON    @75          =
   920 1601           440 CON    @440            BLANK
   921 1602             1 GOSUB  PRTM          PUT OUT REG CONTENT
   921 1603             0
   922 1604             1 GOSUB  EOLL          PRINT THE LINE
   922 1605             0
   923 1606             1 GOSUB  PWAIT
   923 1607             0
   924 1610          1170 C=REGN 9             DONE YET
   925 1611          1072 C=C+1  M
   926 1612          1150 REGN=C 9
   927 1613           246 AC EX  X
   928 1614            74 RCR    3
   929 1615          1406 ? A<C  X
   930 1616             1 GOLNC  REGLOP
   930 1617             2
   931 1620             1 GOSUB  PECHK         CHECK PRINTER ERRORS
   931 1621             0
   932 1622          1110 S9=    1             FOR CARD READER
   933 1623          1740 RTN
*
*******************************************************************
* PRNOP - THIS IS A DUMMY FUNCTION TO MAKE THE FUNCTION NUMBER    *
*         INCREASE TO 33                                          *
*******************************************************************
*
   940                    ENTRY  PRNOP
   941 1624           255 CON    @255         -
   942 1625            55 CON    @55          -
   943 1626 PRNOP  1740 RTN
*
*- A-C= REG A  -  REG C
*
*-SETDEC, SUBTRACT REGS A&C, GO TO "DATA ERROR" FOR OVERFLOW OR UNDERFLOW
*    (DOESN'T MESS WITH RAM)
*
*-USES:   A,B,C,M,      PT,      NO STS ??         1 SUB LEVEL
*-INPUTS:  REG A&C= FLOATING POINT, NORMALIZED NUMBERS
*-OUTPUTS: C= A-C  (FLOATING POINT),    DEC MODE,  PT= 12 -- OK
*                                   PT= 11 -- UNDERFLOW, PT= 10 -- OVERFLOW
*
   955                    ENTRY  A-C
   956 1627 A-C     1240 SETDEC
   957 1630          1276 C=-C-1 S
   958 1631             0 NOP
   959 1632             1 GOSUB  AD2-10        ADD "A" TO "-C"
   959 1633             0
   960 1634             1 GOLONG OVFL10        CHECK FOR OVER/UNDER FLOW
   960 1635             2
*
```

```
*
*******************************************************************
*********** PRT14 -- EXITING FROM ALPHA MODE WITH ALPHA KEY ************
*******************************************************************
   966                     ENTRY  ENDALP
   967 1636 ENDALP   530 M=C                  SAVE REG C
   968 1637            1 GOSUB  DATAPR        PRINT ALPHA ENTRY STRING
   968 1640            0
   969 1641           34 PT=    3
   970 1642          630 C=M                  RESTORE REG C
   971 1643            1 GOLONG PR14RT
   971 1644            2
   972                   FILLTO @1644
*
*******************************************************************
******************  PRT12 -- PRINT CATALOG  *************************
*******************************************************************
   977                   ENTRY  PRTCAT
   978 1645 PRTCAT   404 S8=    0
   979 1646            1 GOSUB  IAUALL
   979 1647            0
   980 1650         1740 RTN                  P+1 - DON'T PRINT
   981 1651         1070 C=REGN 8             GET CATALOG #
   982 1652         1176 C=C-1  S
   983 1653         1176 C=C-1  S             CATALOG 1 ?
   984 1654          313 GONC   DOLCD  (1705) NO
**FOR CAT 1, IF PGM PTR IS AT AN END, PRINT THE NUMBER OF BYTES BETWEEN
*THE PREVIOUS END AND THIS END, INCLUDING 3 BYTES FOR THIS END.
   987 1655            1 GOSUB  GETPC         YES, A(0-3)= PGM POINTER
   987 1656            0
   988 1657            1 GOSUB  INCAD         INCREMENT ADDRESS= 1ST BYTE
   988 1660            0
   989 1661          212 B=A    WPT           SAVE COPY OF 1ST BYTE ADDRESS
   990 1662            1 GOSUB  INCAD         SKIP 2ND BYTE
   990 1663            0
   991 1664            1 GOSUB  NXTBYT        GET 3RD BYTE
   991 1665            0
   992 1666         1530 ST=C                 SAVE 3RD BYTE IN STATUS
   993 1667         1434 PT=    1
   994 1670         1042 C=C+1  PT            ALPHA LBL?
   995 1671          147 GOC    DOLCD  (1705) YES
   996 1672           34 PT=    3             IT'S AN END
   997 1673          252 AC EX  WPT           C= 3RD BYTE ADDRESS
   998 1674          530 M=C                  SAVE 3RD BYTE ADDRESS
   999 1675          214 ?S5=1                FINAL END
  1000 1676          123 GONC   PCAT20 (1710) NO
  1001 1677            1 GOSUB  PR.END        YES. PRINT ".END."
  1001 1700            0
  1002 1701            1 GOSUB  PRTMSG
  1002 1702            0
  1003 1703          647 CON    @647          SKIP 7 CHARACTERS
  1004 1704           63 GOTO   PCAT25 (1712)
  1005 1705 DOLCD      1 GOSUB  PRTLCD
  1005 1706            0
  1006 1707          263 GOTO   OUTPCT (1735)
  1007 1710 PCAT20     1 GOSUB  PRTLCD
  1007 1711            0
  1008 1712 PCAT25    34 PT=    3
  1009 1713          152 AB EX  WPT           A= PC= 1ST BYTE OF END
  1010 1714            1 GOSUB  CPGMHD        A= ADDR OF TOP OF PROGRAM
```

```
 1010 1715             0
 1011 1716           630 C=M
 1012 1717           352 BC EX   WPT          B=3RD BYTE ADDR OF END
 1013 1720             1 GOSUB   CNTBYT                 COUNT # OF BYTES
 1013 1721             0
 1014 1722           246 AC EX   X            C.X= TOTLA # OF BYTES
 1015 1723             1 GOSUB   PBINB0       PRINT # BYTES
 1015 1724             0
 1016 1725             1 GOSUB   PRTMSG
 1016 1726             0
 1017 1727            40 CON     @40                BLANK
 1018 1730           102 CON     @102         B
 1019 1731           131 CON     @131         Y
 1020 1732           124 CON     @124         T
 1021 1733           105 CON     @105         E
 1022 1734           523 CON     @523         S
 1023                    FILLTO @1734
*
* THIS ENTRY IS USED BY TIMER ROM TOO./ IT REQUIRED :
* USED ONLY A,C,N,S0-S7,S9 AND +2 SUB LEVEL
*  63DD
 1028 1735 [redacted]    1 GOSUB   EOLL (67    SEND EOLL
 1026 1736             0
→1029 1737←63DF        1 GOSUB   BECHK(6E3C)  WAIT FOR PRINTER  Buffer Empty Check
 1029 1740             0                      Entry Point from TIMER
3E4 1030 1741          1 GOLONG PECHK (6D27)  Printer Error Check
 1030 1742             2
 1031
*************************************************************************
*
* BECHK (BUFFER EMPTY CHECK) - WAIT UNTIL PRINTER IS IDLE OR PRINTER
* BUFFER IS EMPTY. NOTE THAT WHEN THE PRINTER RUNS OUT OF PAPER, IT
* MAY GO IDLE WHILE THERE IS STILL DATA IN ITS BUFFER.
*
* USES C,NO PT, S7-S0,S9 (ERRORS). LEAVES ORIGINAL S7-S0 IN C[1:0].
*      USES ONE ADDITIONAL SUBROUTINE LEVEL.
*
* INPUT: NONE
* OUTPUT: 1ST BYTE OF PRINTER STATUS IS IN S7-S0.  2ND BYTE OF PRINTER
*      STATUS IS IN C[13:12].
* ASSUMES: S9 IS PRINTER INTERFACE ERROR FLAG.
*
 1046                    ENTRY   BECHK
3E 1047 1743 BECHK     1 GOSUB   FNSTS
 1047 1744             0
 1048 1745 BECK20     14 ?S3=1                ODPS?
 1049 1746            23 GONC    BECK30 (1750) NO
 1050 1747          1110 S9=     1            SET ERROR FLAG
 1051 1750 BECK30   1114 ?S9=1                ANY ERROR?
 1052 1751          1540 RTN C
 1053 1752           776 C=C+C   S
 1054 1753           776 C=C+C   S            IDLE?
 1055 1754          1540 RTN C
 1056 1755           776 C=C+C   S            BUFFER EMPTY?
 1057 1756          1540 RTN C
 1058 1757          1730 CST EX               RESTORE ORIGINAL STATUS
 1059 1760             1 GOSUB   FNSTS
 1059 1761             0
 1060 1762          1633 GOTO    BECK20 (1745)
*************************************************************************
```

```
 *
 * PWAIT (PRINTER WAIT) - WAIT FOR BUFFER EMPTY OR IDLE, THEN CHECK
 * FOR PRINTER ERROR AND CHECK FOR KEYDOWN
 *
 * USES: C,A(X),NO PT, S9 FOR ERRORS,  2 ADDITIONAL SUBROUTINE LEVELS
 *
 * INPUT: NONE
 * OUTPUT: NONE
 * ASSUMES: S9 IS PRINTER INTERFACE ERROR FLAG
 *
   1072                       ENTRY  PWAIT
   1073                       ENTRY  PWAITX
   1074 1763 PWAIT        1 GOSUB  BECHK
   1074 1764              0
   1075 1765          1730 CST EX                 RESTORE ORIGINAL STATUS
   1076 1766 PWAITX 1114 ?S9=1                  ANY ERROR SO FAR ?
   1077 1767              1 GOLC   PEDIAG          GOTO SEE WHAT'S WRONG IF ERROR
   1077 1770              3
   1078 1771              1 GOLONG PCHKKB
   1078 1772              2
*
**********************************************************************
*
*-CLR&SS= CLEAR RUNNING & SST FLAG
*          ALSO CLEARS PAUSING
*
*-USES:     C,  S0-S7,  NO PT,  1 ADDITIONAL SUB LEVEL
*-IN:       NOTHING
*-OUT:      SS0 UP, CHIP 0 ENABLED,   RUNNING,SSTFLAG,&PAUSING CLEARED
*-ASSUMES:  NOTHING
*
  1090                       ENTRY  CLR&SS
  1091 1773 CLR&SS       1 GOSUB  LDSST0         LOAD STATUS SET 0
  1091 1774              0
  1092 1775            104 S4=     0              CLEAR SST FLAG
  1093 1776              1 GOLONG STOPSB         CLEAR PAUSING&RUNNING,
  1093 1777              2
  1094                                              & STORE AWAY SST0
*
  1096                       UNLIST
  1095                       END

 ERRORS :      0
```

```
SYMBOL  TABLE

A-C      1627   -
ACA      1264   -
BECHK    1743   -
BECK20   1745   -   1762
BECK30   1750   -   1746
CLR&SS   1773   -
DOLCD    1705   -   1671 1654
ENDALP   1636   -
ERL      1170   -   1143
LPECHK   1242   -
MORALP   1542   -   1551
MSG      1576   -   1562 1552
OUTPCT   1735   -   1707
PAR40    1304   -   1334 1320
PAR50    1311   -   1306
PAR70    1317   -   1310
PAREG    1271   -
PCAT20   1710   -   1676
PCAT25   1712   -   1704
PPRUMP   1246   -
PRA      1234   -
PRA20    1236   -   1260
PRAXIS    416   -
PRNOP    1626   -
PRPLOT     75   -
PRPLTP    246   -
PRREG    1361   -
PRREGX   1416   -
PRSIGN   1374   -
PRSTK    1342   -
PRSTKX   1347   -
PRTCAT   1645   -
PWAIT    1763   -
PWAITX   1766   -
REG      1553   -   1501
REGL     1444   -   1370
REGL00   1452   -   1353
REGLOP   1454   -
SIGSTP   1532   -
SKPC10   1224   -   1153
SKPC20   1226   -   1221
SKPC4    1201   -   1177
SKPC6    1210   -   1212
SKPC8    1222   -   1207
SKPCHR   1135   -
SKPCOL   1162   -
SKPCOM   1200   -
STK      1563   -   1477
STKADR   1571   -
STKCHK   1447   -   1407
STKCKX   1407   -
XPECHK   1244   -   1270 1230
```

```
ENTRY TABLE

A-C      1627   -
ACA      1264   -
BECHK    1743   -
CLR&SS   1773   -
ENDALP   1636   -
LPECHK   1242   -
PAREG    1271   -
PPROMP   1246   -
PRA      1234   -
PRA20    1236   -
PRAXIS    416   -
PRNOP    1626   -
PRPLOT     75   -
PRPLTP    246   -
PRREG    1361   -
PRREGX   1416   -
PRSIGM   1374   -
PRSTK    1342   -
PRSTKX   1347   -
PRTCAT   1645   -
PWAIT    1763   -
PWAITX   1766   -
REGL     1444   -
REGLOP   1454   -
SIGSTF   1532   -
SKPC4    1201   -
SKPCHR   1135   -
SKPCOL   1162   -
SKPCOM   1200   -
STKADR   1571   -
```

```
EXTERNAL REFERENCES

ACA         5
ACA         4
ACCHR       7
ACCHR       6
ACCOL      11
ACCOL      10
ACSPEC     13
ACSPEC     12
ACX        15
ACX        14
AD2-10   1632
AD2-10   1633
BECHK    1737   1763
BECHK    1740   1764
BLDSPC     17
BLDSPC     16
CHKADR   1460
CHKADR   1461
CKANGL   1311   1543
CKANGL   1312   1544
CKEN     1246
CKEN     1247
CNTBYT   1720
CNTBYT   1721
CONV3C   1433
CONV3C   1434
CONV3D   1135   1162   1416
CONV3D   1136   1163   1417
CPCMHD   1714
CPCMHD   1715
DATAPR   1637
DATAPR   1640
EOLL     1242   1452   1604   1735
EOLL     1243   1453   1605   1736
ERRDE    1170
ERRDE    1171
FMT        65
FMT        64
FNDEND   1361
FNDEND   1362
FNDPTR   1252
FNDPTR   1253
FNSTS    1743   1760
FNSTS    1744   1761
GETPC    1655
GETPC    1656
IACHR    1145   1264
IACHR    1146   1265
IACOL    1173
IACOL    1174
IAUALL   1646
IAUALL   1647
IAUNB    1255
IAUNB    1256
INCAD    1657   1662
INCAD    1660   1663
```

```
 INTFRC  1425
 INTFRC  1426
 IPRT    1234  1342  1450
 IPRT    1235  1343  1451
 LDSST0  1773
 LDSST0  1774
 LIST      21
 LIST      20
 LISTEN  1465
 LISTEN  1466
 NXTBYT  1664
 NXTBYT  1665
 OVFL10  1634
 OVFL10  1635
 PAREG   1236  1266
 PAREG   1237  1267
 PBINB0  1560  1723
 PBINB0  1561  1724
 PBYA+C  1224
 PBYA+C  1225
 PBYTDU  1313
 PBYTDU  1314
 PBYTEC  1214  1545  1555  1574
 PBYTEC  1215  1546  1556  1575
 PCHKKB  1771
 PCHKKB  1772
 PECHK   1244  1620  1741
 PECHK   1245  1621  1742
 PEDIAG  1767
 PEDIAG  1770
 PHEAD      3
 PHEAD      2
 PR.END  1677
 PR.END  1700
 PR14RT  1643
 PR14RT  1644
 PRA       23
 PRA       22
 PRAXIS    25
 PRAXIS    24
 PRBUF     27
 PRBUF     26
 PRFLAG    31
 PRFLAG    30
 PRKEYS    33
 PRKEYS    32
 PRNOP     67
 PRNOP     66
 PRP       35
 PRP       34
 PRPLOT    37
 PRPLOT    36
 PRPLTP    41
 PRPLTP    40
 PRREG     43
 PRREG     42
 PRREGX    45
 PRREGX    44
 PRSIGM    47
 PRSIGM    46
```

```
PRSTK      51
PRSTK      50
PRTLCD   1705   1710
PRTLCD   1706   1711
PRTM     1602
PRTM     1603
PRTMSG   1576   1701   ·1725
PRTMSG   1577   1702   1726
PRX        53
PRX        52
PWAIT    1606
PWAIT    1607
REGLOP   1616
REGLOP   1617
REGPLT     55
REGPLT     54
SIGSTF   1502
SIGSTF   1503
SKPCHR     57
SKPCHR     56
SKPCOL     61
SKPCOL     60
STKADR   1563
STKADR   1564
STKPLT     63
STKPLT     62
STOPSB   1776
STOPSB   1777
SUMCHK   1374
SUMCHK   1375
UNL      1454
UNL      1455

End of VASM assembly
*******************************************************************
VASM ROM ASSEMBLY              REV.  6/81A

OPTIONS: L C S

     2                     FILE   SCPR2B
*
*
*
*
*
**********************************************************************
*
* [redacted]- PRINT MESSAGE. SENDS A LIST OF CONSTANTS (FOLLOWING THE
* "GOSUB PRTMSG") TO THE PRINTER, STOPPING WHEN IT SEES THE 9TH BIT=1,
* USES THE CPBYTE OUTPUT SUBROUTINE, SO OUTPUT IS CONDITIONED ON
* FLAG 55.  IF THE 10TH BIT=1 IT WAITS FOR BUFFER EMPTY, THEN CHECKS
* FOR PRINTER ERRORS, AND THEN CHECKS FOR "R/S" AND "ON" KEYS, BEFORE
* CONTINUING TO PRINT THE LIST OF CONSTANTS.  THE 9TH AND 10TH BITS
* MAY NOT BOTH BE SET IN THE SAME CONSTANT.  WHEN THE 10TH BIT IS SET,
* IF "R/S" OR "ON" IS DOWN OR AN ERROR HAS OCCURRED, PRTMSG ABORTS.
*
* USES: FOR BIT 10=0: C,N, NO PT, S9, HEXMODE, 1 ADDITIONAL SUBROUTINE
*                     LEVEL
*       FOR BIT 10=1: A,X, C, N,  NO PT,  S9,  ? ADDITIONAL SUB LEVELS
*                     NOTE THESE BIT 10=1 COMMENTS ARE PARTLY GUESSES.
* IN: LIST OF CONSTANTS FOLLOWING THE "GOSUB PRTMSG", WHERE THE LAST
*      CONSTANT HAS THE 9TH BIT=1 TO FLAG THE END OF THE LIST.
* OUT: MESSAGE TO PRINTER (IF FLAG 55=1), CHIP 0 ENABLED, HEXMODE,
*      S9=1 FOR ERRORS.
* ASSUMES: HEXMODE
*
* PRTMSL - SAME AS PRTMSG EXCEPT WILL OUTPUT AN EOLL IF LAST EOL
*          IS NOT A EOLL
*
*CAUTION!!!  DO NOT MOVE PRTMSG FROM THIS LOCATION (QUAD 2, @000) !!!!
*            IT MAY BE USED BY OTHER PLUG-IN ROMS.

                    ENTRY   PRTMSG
         PRTMSG     ENTRY   PRTMSL
6400   0 [redacted] 660 C=STK                     GET ADDR OF 1ST CHAR
       1 PRTMS1 1460 CXISA                        GET CHAR
       2           1 GOSUB  CPBYTE (6E4P)         SEND CHAR TO PRINTER
       3           0
       4        1072 C=C+1  M                     INC ADDR
       5        1366 ? C#0  XS                    DONE?
       6        1733 GONC   PRTMS1 (    1) NO
       7         560 STK=C                        PUT CHR POINTER ON STK
      10         766 C=C+C  XS                    IS THIS A 1000 CODE?
      11         766 C=C+C  XS
      12         766 C=C+C  XS
      13        1640 RTN NC                       NO A 400 CODE
      14           1 GOSUB  PWAIT                 WAIT FOR THE PRINTER
      15           0
      16        1623 GOTO   PRTMSG (    0)
*
640F  17 PRTMSL  644 C=HPIL 6                     GET LAST STATUS
      20         672
      21         603
      22        1474 RCR    1
      23         776 C=C+C  S                     LAST EOL AN EOLL ?
      24        1543 GONC   PRTMSG (    0) YES
      25         460 LDI
      26         340 CON    @340   EØ
      27           1 GOSUB  CPBYTE                SEND AN EOLL
      30           0
      31        1473 GOTO   PRTMSG (    0)
*****************************************************************************
                    ENTRY   OVERFL
      32 OVERFL 1140 SETHEX
      33           1 GOSUB  IAUNA                 OK TO PRINT?
      34           0
      35        1740 RTN                          P+1 -- DON'T PRINT
      36           1 GOSUB  ACXSUB                P+2 -- PRINT X REGISTER
      37           0
      40         373 GOTO   DATP25 (   77)

*OVERFL FALLS INTO DATAPR HERE!!!!!!!!!!!!!!!!!!!!

                    EJECT
```

```
* DATAPR - PRINT DATA ENTRY STRING AND CLEAR DATAENTRY FLAG
* IF PRINTER IS OFF OR IN MANUAL MODE, RETURNS WITHOUT PRINTING.
* IF ANY PRINTER ERROR, CALLS RSTSEQ AND GOES TO PEDIAG (NEVER
* RETURNS).
*
* REQUIRES CHIP 0 SELECTED ON ENTRY
* DOES NOT REQUIRE HEXMODE OR P SELECTED ON ENTRY
* USES 3 ADDITIONAL SUBROUTINE LEVELS!
* USES A, B, C, G, N, P, Q, S0-S9
* LEAVES HEXMODE, CHIP 0 SELECTED, P SELECTED
* PRESERVES M
*
                  ENTRY  DATAPR
  41 DATAPR 1140 SETHEX
  42         240 SEL P
  43        1670 C=REGN 14
  44        1074 RCR    2
  45        1530 ST=C                     PUT UP SS1
  46        1014 ?S2=1                    DATAENTRY FLAG SET?
  47        1640 RTN NC                   NOPE
  50        1004 S2=    0                 CLEAR DATAENTRY FLAG
  51        1630 C=ST
  52        1574 RCR    12
  53        1650 REGN=C 14                PUT SS1 BACK

  54           1 GOSUB  IAUNA
  55           0
  56        1740 RTN                      P+1 - DON'T PRINT
                                          P+2 - OK TO PRINT
  57        1670 C=REGN 14                RESTORE SS0
  60        1530 ST=C
  61          14 ?S3=1                    PROGRAM MODE?
  62          43 GONC   DATP15 (   66)    NO
* WE'RE IN PROGRAM MODE WITH THE DATA ENTRY FLAG SET.  A DIGIT ENTRY
* STRING OR ALPHA ENTRY STRING HAS JUST BEEN INSERTED INTO PROGRAM
* MEMORY.  LINE# MUST BE VALID AND NON-ZERO.  PRIVACY MUST BE CLEAR.
  63           1 GOSUB  PPGMST
  64           0
  65          53 GOTO   DATP17 (   72)

  66 DATP15 1214 ?S7=1                    ALPHAMODE?
  67          63 GONC   DATP20 (   75)    NO
                                          YES, ALPHAMODE
  70           1 GOSUB  PAREG             SEND ALPHA REG TO PRINTER
  71           0
  72 DATP17    1 GOSUB  EOLL
  73           0
  74         103 GOTO   DATP30 (  104)

  75 DATP20    1 GOSUB  PRTDEF            PRINT FORMATTED STRING
  76           0
                 ENTRY  DATP25            FOR PRT5
  77 DATP25    1 GOSUB  PRTMSG
 100           0
 101         647 CON    @647              SKIP 7 CHARACTERS
 102           1 GOSUB  EOLR              EOLR
 103           0
                 ENTRY  DATP30            USED BY PRT5
 104 DATP30 1114 ?S9=1                    ANY ERROR ?
```

```
   122  105              1 GOLNC  UNL            NO, SEND UNLISTEN
   122  106              2
   123  107              1 GOSUB  RSTSEQ
   123  110              0
   124  111              1 GOLONG PEDIAG
   124  112              2
*****************************************************************************
*-RG9PRT= REG 9 TO PRINTER
*
*-PUT D.E. STRING IN REG 9 INTO SAME FORMAT AS OUTPUT BY "FORMAT"
* (PLEASE REFER TO DIGENT (CN2, @66) FOR FORMAT OF INPUT D.E. STRING)
* S0 - D.P. HIT                 S1 - EEX HIT
* S2 - CHS HIT                  S4 - DIGIT GROUPING FLAG
* S5 - DECIMAL POINT FLAG
*
*
* PDIGE - PRINT DIGIT ENTRY STRING.  ENTRY POINT FOR PRT5 LOGIC
*
   137                    ENTRY  PDIGE
   138  113 PDIGE      1 GOSUB  INIT5
   138  114            0
   139                    ENTRY  PRTDEF
   140  115 PRTDEF 1070 C=REGN 8            LOAD FLAGS - S2:CHS
   141  116         674 RCR    11
   142  117        1530 ST=C                  S1 : EEX   S0:D.P.
   143  120           4 S3=    0            CLEAR LEADING D.P. FLAG
   144  121        1170 C=REGN 9
   145  122         416 A=C    W            A _ REG.9
   146  123        1670 C=REGN 14           GET # TRAILING DIGITS
   147  124        1074 RCR    2
   148  125         366 BC EX  XS           # TRAILING DIGITS TO B(XS)
   149  126           1 GOSUB  LOAD3        LOAD ALL 3'S INTO C
   149  127           0
   150  130          34 PT=    3            START FROM END OF MANTISSA
   151  131          43 GOTO   RG9P13 ( 135)
   152  132 RG9P10 1142 C=C-1  PT           C(PT) _ 2
   153  133         676 A=A-1  S            DECREMENT D.P. POS COUNTER
   154  134        1734 INC PT              POINT TO LEFT NEXT DIGIT
   155  135 RG9P13  542 A=A+1  PT           FOUND THE LAST DIGIT?
   156  136        1747 GOC    RG9P10 ( 132) NO
   157  137         642 A=A-1  PT           YES, RESTORE THE DIGIT
   158  140        1614 ?S0=1               D.P. HIT ?
   159  141         133 GONC   RG9P20 ( 154) NO, DON'T LOOK FOR D.P.
   160  142          23 GOTO   RG9P19 ( 144) YES, LOOK FOR D.P.
   161  143 RG9P17 1734 INC PT              POINT TO LEFT NEXT DIGIT
   162  144 RG9P19  676 A=A-1  S            FOUND THE D.P.?
   163  145        1763 GONC   RG9P17 ( 143) NO
   164  146           1 GOSUB  LDDP10       YES, LOAD THE D.P./COMMA
   164  147           0
   165  150         242 AC EX  PT           D.P./COMMA BACK TO "C"
   166  151        1324 ?PT=   13           LEADING D.P.?
   167  152          23 GONC   RG9P20 ( 154) NO
   168  153          10 S3=    1            YES, SET LEADING D.P. FLAG
   169  154 RG9P20  114 ?S4=1               GROUPING FLAG SET ?
   170  155         263 GONC   RG9P29 ( 203) NO
   171  156         340 SEL Q               YES
   172  157        1034 PT=    2
   173  160 RG9P24 1734 INC PT              LOOK FOR P
   174  161         440 ?P=Q                FOUND P?
   175  162        1763 GONC   RG9P24 ( 160) NO
```

```
163            1324 ? PT=   13              YES, NOW P=Q
164             217 GOC     RG9P30 ( 205)
165 RG9P26      436 A=C     S               A(13) _ 3
166 RG9P27      676 A=A-1   S               COUNT 3 FROM LEFT
167              57 GOC     RG9P28 ( 174)  PUT A COMMA HERE ?
170            1524 ? PT=   12              NO,REACH LEFT END OF MANTISSA ?
171             147 GOC     RG9P30 ( 205)  YES, DONE
172            1734 INC PT                  POINT TO LEFT NEXT DIGIT
                    LEGAL
173            1733 GOTO    RG9P27 ( 166)
174 RG9P28      214 ?S5=1                   LOAD A COMMA TO C
175              33 GONC    *+3     ( 200)
176            1720 LC      15
177              23 GOTO    *+2     ( 201)
200             720 LC      7               LOAD A D.P. INSTEAD OF
201            1734 INC PT                  RESTORE POINTER
                    LEGAL
202            1633 GOTO    RG9P26 ( 165)
203 RG9P29     1326 ? B#0   XS              DISPLAY MODE= 0?
204             233 GONC    RG9P35 ( 227)  YES, NO TRAILING ZEROS
205 RG9P30      240 SEL P
206            1214 ?S7=1                   FIX MODE?
207             203 GONC    RG9P35 ( 227)  NO, NO TRAILING ZEROS
210            1414 ?S1=1                   YES, EEX HIT?
211             167 GOC     RG9P35 ( 227)  YES, NO TRAILING ZEROS
212               1 GOSUB   LDDP10          NO, LOAD D.P./COMMA
213               0
214             242 AC EX   PT              D.P./COMMA BACK TO "C"
215             366 CB EX   XS              # TRAILING DIGITS TO "C"
216            1724 DEC PT                  PT TO 1ST TRAILING DIGIT
                    LEGAL
217              43 GOTO    RG9P33 ( 223)
220 RG9P32      320 LC      3               ADD TRAILING DIGIT
221-           1024 ?PT=    2               REACHED END OF MANTISSA?
222              47 GOC     RG9P34 ( 226)  YES
223 RG9P33     1166 C=C-1   XS              NO, COUNT TRAILING DIGIT
224            1743 GONC    RG9P32 ( 220)
225            1034 PT=     2
226 RG9P34      320 LC      3               RESTORE C(XS)
227 RG9P35      436 A=C     S               TAKE CARE OF THE SIGN
230             676 A=A-1   S               A(13) _ 2
231             136 C=0     S               ASSUME POSITIVE MANTISSA
232            1334 PT=     13
233            1014 ?S2=1                   CHS HIT ?
234              23 GONC    *+2     ( 236) NO, MANTISSA POSITIVE
235            1520 LC      13              "-" = 2D
236             276 AC EX   S
237            1166 C=C-1   XS              C(2) _ 2
240            1414 ?S1=1                   EEX HIT ?
241             213 GONC    RG9P50 ( 262)  NO, DONE
242            1046 C=C+1   X               YES, C(0)= 3
243            1434 PT=     1               LOOK AT DIGIT 1
244             542 A=A+1   PT              IS THERE A DIGIT THERE ?
245             127 GOC     RG9P42 ( 257)  NO, EXP = 00
246             642 A=A-1   PT              YES, RESTORE DIGIT 1
247            1634 PT=     0               LOOK AT DIGIT 0
250             542 A=A+1   PT              IS THERE A DIGIT ?
251              43 GONC    RG9P40 ( 255)  YES
252            1434 PT=     1               NO
253            1612 A SR    WPT             MAKE 2D EXP
```

```
254                 43 GOTO    RG9P45 ( 260)
255 RG9P40         642 A=A-1   PT                RESTORE DIGIT 0
                       LEGAL
256                 23 GOTO    RG9P45 ( 260)
257 RG9P42          12 A=0     WPT
260 RG9P45          34 PT=     3                 SAY PRINT EXP
261                 33 GOTO    OUTRG9 ( 264)
262 RG9P50          26 A=0     XS
263               1634 PT=     0                 SAY ONLY PRINT MANTISSA
264 OUTRG9         723 GOTO    PDIGAC ( 356)
                       EJECT
```

```
**********************************************************************
**********************  PRT 10= VIEW  ********************************
**********************************************************************
                               ENTRY  PVIEW
54B5    265 PVIEW    116 C=0                    RE-ENABLE CHIP 0
        266         1160 DADD=C
        267            1 GOSUB  CKEN            OK TO PRINT ?
        270            0
        271         1740 RTN                    P+1 - NO
        272            1 GOSUB  FNDPTR          P+2 - YES, SEE IF PTR THERE
        273            0
        274          153 GOTO   PVW10  ( 311) NO PRINTER
        275            1 GOSUB  INITC
        276            0
        277           40 SPOPND                 SAVE A SUBR LEVEL
        300          316 C=B                    SAVE VALUE TO BE VIEWED
        301          530 M=C                     IN M
        302            1 GOSUB  ACREGC
        303            0
        304            1 GOSUB  RPECHK          EOLR, CHECK PRINTER ERRORS
        305            0
        306          630 C=M                    RESTORE VALUE TO C
        307            1 GOLONG PRIORT
        310            2
        311 PVW10   1304 S13=    0
        312         1740 RTN
*
* ACXSUB (SUBROUTINE TO ACCUMULATE X) - SENDS WHATS IN THE X REGISTER
* TO THE PRINTER BUFFER
* USES: A,B,C,N,P,Q,G,S0-S9 AND 2 ADDITIONAL SUBROUTINE LEVELS
* CAUTION: I'M GUESSING AT WHAT FORMAT AND PDIGAC USE WHEN THEY ARE
*      CALLED BY ACXSUB
* INPUTS: GETS VALUE OF X FROM R3
* OUTPUTS: A CHARACTER STREAM TO THE PRINTER BUFFER
* ASSUMES: CHIP 0 ENABLED, S9 IS THE PRINTER INTERFACE ERROR FLAG
*      HEXMODE
*
* ACREGC (ACCUMULATE C REGISTER) - SAME AS ACXSUB EXCEPT ASSUMES INPUT
* VALUE IS IN C ON ENTRY.
*
* PRTM - SAME AS ACXSUB EXCEPT ASSUMES INPUT VALUE IN M ON ENTRY
*
                                ENTRY  PRTM
        313 PRTM      630 C=M
        314            23 GOTO   ACREGC ( 316)
                                ENTRY  ACREGC
                                ENTRY  ACXSUB
64CD    315 ACXSUB    370 C=REGN 3 (X)
64CE    316 ACREGC     36 A=0     MS
        317           576 A=A+1   MS
        320          1576 ? A#C   MS             NUMERIC DATA?
        321            63 GONC   ALPDAT ( 327) NO, ALPHA DATA
        322             1 GOSUB  FORMAT         YES, FORMAT THE NUMBER
                                 CA95
        323             0
64D4    324           156 AB EX
        325             4 S3=    0              NO LEADING D.P.
        326           323 GOTO   PDIGAB ( 360) SEND NUMBER TO PRINTER
        327 ALPDAT    416 A=C                   SAVE C
        330             1 GOSUB  PRQUOT
```

```
331                 0
332               256 AC EX                     RESTORE C
333              1574 RCR     12
334              1434 PT=     1
335               112 C=0     WPT
336              1356 ?C#0                      ANY ALPHA DATA?
337               133 GONC    ALPD55 ( 352) NO, ALL NULLS
340 ALPD45       1574 RCR     12                CHAR TO C( 0-1 )
341              1352 ?C#0    WPT               NULL?
342              1763 GONC    ALPD45 ( 340) YES, GET NEXT CHAR
343 ALPD50          1 GOSUB   CKANGL            CHECK IF THE CHAR IS AN ANGEL SIGN
344                 0
345                 1 GOSUB   PBYTDU
346                 0
347              1574 RCR     12                NEXT CHAR TO C( 0-1 )
350              1352 ?C#0    WPT               NULL?
351              1727 GOC     ALPD50 ( 343) NO
    ALPD55
                      ENTRY   PRQUOT
352 PRQUOT        460 LDI
353                42 CON     @42               QUOTATION MARK
354                 1 GOLONG CPBYTE
355                 2
***********************************************************************
*-INPUTS: [PDIGAB]  B= DIGITS,    A= PUNCTUATION
*         [PDIGAC]  A= DIGITS,    C= PUNCTUATION
*     BOTH ENTRIES: P SELECTED, HEX MODE
*
*-USES:  A,B,C,G,N,P,Q,    S3, S9 FOR ERRORS, 1 ADDITIONAL SUB LEVEL
*-OUTPUTS: HEX MODE,  DOESN'T USE OR CHANGE CHIP ENABLE
*
                      ENTRY   PDIGAB
                      ENTRY   PDIGAC
356 PDIGAC        216 B=A                       DIGITS TO "B"
357               416 A=C                       PUNCTUATION TO "A"
360 PDIGAB        460 LDI                                              6-F2
361              1000 CON     @1000 [220]
362              1624 ?PT=    0                 PRINT EXPONENT?
363                23 GONC    PDIG10 ( 365) YES
364               406 A=C     X                 NO,A( 0-1 )=0=FLAG, A(XS)= BLANK
365 PDIG10       1074 RCR     2                 C( 0 )= 2
366               336 C=B     S                 GET SIGN OF NUMBER
367              1374 RCR     13                PUT IT IN C( 0-1 )
370                 1 GOSUB   PBYTEC            SEND BLANK OR "-" TO PRINTER
371                 0         6E22
372               460 LDI
373                56 CON     @56 [0            ASCII D.P.
374                14 ?S3=1                     PRINT LEADING D.P.?
375                 1 GSUBC   PBYTEC            YES, D.P. TO PRINTER
376                 1
377              1534 PT=     12
400 PDIG25        320 LC      3
401              1734 INC PT
402              1402 ?A<C    PT                BLANK?
403               143 GONC    PDIG30 ( 417) NO
404              1434 PT=     1                 YES
405              1512 ? A#0   WPT               EXPONENT NEEDED?
406              1640 RTN NC                    NO, FIX MODE
407              1034 PT=     2                 YES
410              1326 ? B#0   XS                EXPONENT POSITIVE?
```

```
411               47 GOC     PDIGXS ( 415) NO, NEGATIVE
412             1320 LC      11            YES, POSITIVE
413             1034 PT=     2
414              342 BC EX   PT            FIX "B" TO PUT OUT A "+"
415 PDIGXS       220 LC      2
416             1034 PT=     2
417 PDIG30      1374 RCR     13
420              342 CB EX   PT            DIGIT TO "C"
421              130 G=C                   ASCII DIGIT TO "G"
422              340 SEL Q
423             1634 PT=     0
424              230 C=G                   DIGIT TO C(0-1)
425                1 GOSUB   PBYTEC        SEND BYTE TO PRINTER
426                0
427              240 SEL P
430             1474 RCR     1             MOVE THE "3" BACK TO C(PT)
431             1542 ? A#C   PT            PUNCTUATION?
432              123 GONC    PDIG50 ( 444) NO
433              460 LDI
434               54 CON     @54           ASCII COMMA
435              242 AC EX   PT            PUNCTUATION TO "C"
436              742 C=C+C   PT            COMMA?
437               37 GOC     PDIG48 ( 442) YES
440             1046 C=C+1   X             NO, D.P.
441             1046 C=C+1   X             C(X)= @56= ASCII D.P.
                     LEGAL
442 PDIG48         1 GOSUB   PBYTEC        SEND PUNCTUATION TO PRINTER
443                0
444 PDIG50      1724 DEC PT
445             1324 ? PT=   13            DONE?
446             1323 GONC    PDIG25 ( 400) NO
447             1740 RTN                   YES, DONE

                     EJECT
```

```
***********************************************************************
**************** PRT2 -- NEXT INSTRUCTION IN MAIN LOOP  **************
***********************************************************************
                    ENTRY   NXINST
6522  450 NXINST   314 ?S10=1                    ROMFLAG ?
      451         1540 RTN C                      YES
      452          106 C=0     X                 RE-ENABLE CHIP 0
      453         1160 DADD=C
      454         1630 C=ST                      ST TO C[1:0]
      455          414 ?S8=1                     COPY S8 TO C.XS
      456           23 GONC    NXIN10 ( 460)
      457         1066 C=C+1   XS
      460 NXIN10  1150 REGN=C  9                 SAVE MISC INFO IN REG 9
      461         1574 RCR     12                FC TO C[1:0]
      462          126 C=0     XS
      463         1346 ? C#0   X                 IS THIS NON-NULL
      464            1 GOLNC   RUNING            NULL
      465            2
      466            1 GOSUB   CKTRCE            SEE IF PTR IN TRACE MODE
      467            0
      470          113 GOTO    NXIN15 ( 501) NO
      471            1 GOSUB   FNDPTR            LOOK FOR PTR IN LOOP
      472            0
      473           63 GOTO    NXIN15 ( 501) PRINTER NOT FOUND
      474          160 N=C                       SAVE C IN N FOR INITC
      475          114 ?S4=1                     "ALL" MODE?
      476           67 GOC     NXIN21 ( 504) YES
      477            1 GOSUB   UNL
      500            0
      501 NXIN15  1170 C=REGN  9                 RESTORE C-REG
      502         1530 ST=C                      RESTORE STATUS
      503         1740 RTN

* WE ARE SAVING IN R9: R9[13:10]=ORIG C[13:10]
*                      R9.XS=S8
*                      R9[1:0]=S7-0
*
      504 NXIN21     1 GOSUB   GETPCA            GET ORIGINAL PC
      505            0
      506         1270 C=REGN  10
      507          252 C=A     WPT               COPY ORIGINAL PC TO "C"
      510          412
      511         1250 REGN=C  10                SAVE ORIG PC IN R10(3:0)
      512            1 GOSUB   PUTPCD            DECREMENT & STORE PC
      513            0
      514            1 GOSUB   FLINKA            RECOMPUTE PRIVACY
      515            0
      516          116 C=0
      517         1160 DADD=C                    RE-ENABLE CHIP 0
      520         1514 ?S12=1                    PRIVATE?
      521           73 GONC    NXIN30 ( 530) NO
      522            1 GOSUB   UNL
      523            0
      524            1 GOSUB   CLR&SS            YES,CLEAR RUNNING & SSTING
      525            0
      526            1 GOLONG  ERRPR
      527            2

      530 NXIN30   260 C=N                       RESTORE C
```

```
531              1 GOSUB  INITC         INITIALIZE
532              0
533           1270 C=REGN 10            FETCH ORIGINAL PC
534            416 A=C                  PC TO A(3:0)
535           1170 C=REGN 9             GET FUNCTION CODE
536           1574 RCR    12            FC TO C(0-1)
537              1 GOSUB  LBLCK         CHECK FOR LBL
540              0
541            106 C=0    X             RE-ENABLE CHIP 0
542           1160 DADD=C
543            114 ?S4=1                FC= LBL?
544              1 GSUBC  GLINE#        YES,COMPUTE LINE #
545              1
546 NXIN70       1 GOSUB  FNSTS         FETCH PRINTER STATUS
547              0
550           1114 ?S9=1                ERROR?
551            107 GOC    NXIN80 ( 561) YES
552             14 ?S3=1                OOPS?
553             33 GONC   NXIN75 ( 556) NO
554           1110 S9=    1             SET ERROR FLAG
555             43 GOTO   NXIN80 ( 561)
556 NXIN75     776 C=C+C  S
557            776 C=C+C  S             IDLE?
560           1663 GONC   NXIN70 ( 546) NO, WAIT SOME MORE

561 NXIN80    1204 S7=    0             SET UP FOR PPGSNL
562              1 GOSUB  PPGSNL        PRINT PROGRAM STEP
563              0
564              1 GOSUB  EOLR          PRINT RIGHT JUSTIFIED
565              0
566           1114 ?S9=1                ANY PRINTER ERRORS?
567             53 GONC   NXIN90 ( 574) NO
570              1 GOSUB  CLR&SS        CLEAR RUNNING, SST, PAUSING
571              0
572              1 GOLONG PEDIAG
573              2
574 NXIN90     132 C=0    M             PUT NFRPU BACK ON THE
575            134 PT=    4               RTN STACK
576           1720 LC     15            NFRPU= 00F0
*THE "LC" LEAVES PT= 3 !!!!!!!!!
577            560 STK=C
600           1270 C=REGN 10            FETCH ORIGINAL PC
601            412 A=C    WPT           PC TO "A"
602              1 GOSUB  PUTPCF        STORE PC & SET LINE#= FFF
603              0
*
*
604              1 GOSUB  UNL           UNLISTEN
605              0
606           1170 C=REGN 9             RESTORE "C"
607           1530 ST=C                 RESTORE ST
610            404 S8=    0
611           1366 ? C#0  XS            TEST STORED STATUS OF S8
612             23 GONC   NXIN99 ( 614)
613            410 S8=    1
614 NXIN99       1 GOLONG NOPRT         BACK TO MAINFRAME
615              2
*
*
******************************************************************
```

```
531           1 GOSUB  INITC         INITIALIZE
532           0
533        1270 C=REGN 10            FETCH ORIGINAL PC
534         416 A=C                  PC TO A(3:0)
535        1170 C=REGN 9             GET FUNCTION CODE
536        1574 RCR    12            FC TO C(0-1)
537           1 GOSUB  LBLCK         CHECK FOR LBL
540           0
541         106 C=0    X             RE-ENABLE CHIP 0
542        1160 DADD=C
543         114 ?S4=1                FC= LBL?
544           1 GSUBC  GLINE#        YES,COMPUTE LINE #
545           1
546 NXIN70    1 GOSUB  FNSTS         FETCH PRINTER STATUS
547           0
550        1114 ?S9=1                ERROR?
551         107 GOC    NXIN80 ( 561) YES
552          14 ?S3=1                OOPS?
553          33 GONC   NXIN75 ( 556) NO
554        1110 S9=    1             SET ERROR FLAG
555          43 GOTO   NXIN80 ( 561)
556 NXIN75  776 C=C+C  S
557         776 C=C+C  S             IDLE?
560        1663 GONC   NXIN70 ( 546) NO, WAIT SOME MORE

561 NXIN80 1204 S7=    0             SET UP FOR PPGSNL
562           1 GOSUB  PPGSNL        PRINT PROGRAM STEP
563           0
564           1 GOSUB  EOLR          PRINT RIGHT JUSTIFIED
565           0
566        1114 ?S9=1                ANY PRINTER ERRORS?
567          53 GONC   NXIN90 ( 574) NO
570           1 GOSUB  CLR&SS        CLEAR RUNNING, SST, PAUSING
571           0
572           1 GOLONG PEDIAG
573           2
574 NXIN90  132 C=0    M             PUT NFRPU BACK ON THE
575         134 PT=    4               RTN STACK
576        1720 LC     15            NFRPU= 00F0
*THE "LC" LEAVES PT= 3 !!!!!!!!!
577         560 STK=C
600        1270 C=REGN 10            FETCH ORIGINAL PC
601         412 A=C    WPT           PC TO "A"
602           1 GOSUB  PUTPCF        STORE PC & SET LINE#= FFF
603           0
*
*
604           1 GOSUB  UNL           UNLISTEN
605           0
606        1170 C=REGN 9             RESTORE "C"
607        1530 ST=C                 RESTORE ST
610         404 S8=    0
611        1366 ? C#0  XS            TEST STORED STATUS OF S8
612          23 GONC   NXIN99 ( 614)
613         410 S8=    1
614 NXIN99    1 GOLONG NOPRT         BACK TO MAINFRAME
615           2
*
*
**********************************************************************
```

```
652             263 GONC    DF900X ( 700)
653 DF15       1630 C=ST                       SAVE PRINTER STATUS
654             356 BC EX                       IN B[1:0] AND [13:12]
655            1670 C=REGN 14                  PUT UP SS0
656            1530 ST=C
657              14 ?S3=1                      PROGRAM MODE?
660               1 GOLC    DF400              YES
661               3
662            1474 RCR     1                  PUT UP SS 1/2
663            1530 ST=C
664             630 C=M
665             274 RCR     5                  FC TO C3:0
666              34 PT=     3
667             412 A=C     WPT                FC TO A3:0
670            1220 LC      10
671             720 LC      7
672             520 LC      5
673             420 LC      4                  FC FOR PRX=A754
674              34 PT=     3
675            1552 ? A#C   WPT                FC#PRX?
676             157 GOC     DF20   ( 713)
* PRX
* IF THE FCN IS PRX AND THE DATA ENTRY FLAG IS NOT SET, THEN WE DON'T
* PRINT ANYTHING HERE IN PRT5.  WE JUST LET THE PRX FUNCTION ITSELF
* PRINT THE VALUE OF X.
* IF, ON THE OTHER HAND, THE DATA ENTRY FLAG IS SET, THEN PRT5 PRINTS
* THE DIGIT ENTRY STRING AND ABORTS THE PRX FUNCTION.
677             514 ?S6=1                      DATA ENTRY FLAG?
700 DF900X      313 GONC    DF900T ( 731) NO
701               1 GOSUB   PDIGE              PRINT DIGIT ENTRY STRING
702               0
703               1 GOSUB   DATP25
704               0
705               1 GOSUB   RSTSEQ
706               0
707               1 GOLONG NFRPU
710               2
711 DF05J         1 GOLONG DF905
712               2

713 DF20        514 ?S6=1                      DATA ENTRY FLAG?
714             653 GONC    DF200  (1001) NO
715              14 ?S3=1                      ALPHAMODE?
716              67 GOC     DF40   ( 724) YES
717               1 GOSUB   PDIGE              PRINT DIGIT ENTRY STRING
720               0
721             460 LDI
722              21 CON     17                 RIGHT EDGE OF DE STRING
                                               IN CHAR POS 17
723             153 GOTO    DF50   ( 740)

724 DF40       1434 PT=     1                  FC FOR PRA=A748
725             420 LC      4
726            1020 LC      8
727              34 PT=     3
730            1552 ? A#C   WPT                FC#PRA?
731 DF900T      613 GONC    DF900Z (1012) PRA
* THE FUNCTION PRA WILL PRINT THE ALPHA REG, SO THERE'S NO POINT
* IN PRINTING IT HERE.
732               1 GOSUB   INIT5
```

```
 733                 0
 734                 1 GOSUB  PAREG              PRINT ALPHA REG
 735                 0
 736               272 AC EX  M
 737                74 RCR    3                  CHAR COUNT TO C.X
 740 DF50         1634 PT=    0
 741               130 G=C                       SAVE CHAR COUNT IN G
 742                 1 GOSUB  NPFTST             NON-PRINTING FCN?
 743                 0
 744               323 GOTO   DF70    ( 776) P+1 - NON-PRINTING
                                             P+2 - PRINTING
 745              1670 C=REGN 14                 CLEAR FLAG 55 TO SUPPRESS
 746              1156 C=C-1                      PRINTING WHILE
 747              1650 REGN=C 14                   COUNTING CHARACTERS
 750                 1 GOSUB  CPFKB              COUNT CHARS IN FCN DESC
 751                 0
 752                74 RCR    3
 753               406 A=C    X                  SAVE FCN DESC LENGTH IN A.X
 754              1670 C=REGN 14                 RESTORE FLAG 55
 755              1056 C=C+1                      FLAG 55 IS THE
 756              1650 REGN=C 14                   PRINTER EXISTENCE FLAG
 757              1634 PT=    0
 760               230 C=G                       RECOVER ORIGINAL CHAR COUNT
 761               126 C=0    XS
 762               506 A=A+C  X                  A.X=CHAR CT + FCN DESC LENGTH
 763               460 LDI
 764                27 CON    23
 765               246 AC EX  X
 766               706 A=A-C  X                  A.X=23-(CHAR CT+FCN DESC LENGTH)
 767                47 GOC    DF60    ( 773) TOO MUCH FOR ONE LINE
 770                 1 GOSUB  PAD1+A             MAKE FCN DESC RIGHT JUSTIFIED
 771                 0
 772               143 GOTO   DF300   (1006)

 773 DF60            1 GOSUB  FILLIN
 774                 0
 775               113 GOTO   DF300   (1006)

 776 DF70            1 GOSUB  FILLNP
 777                 0
1000 DF900Y        123 GOTO   DF900Z  (1012)

1001 DF200           1 GOSUB  NPFTST
1002                 0
1003                73 GOTO   DF900Z  (1012) P+1 - NON-PRINTING
1004                 1 GOSUB  INIT5              P+2 - PRINTING
1005                 0

     DF300                                   SEND FCN DESC
1006                 1 GOSUB  CPFKB
1007                 0
1010                 1 GOSUB  EOLR
1011                 0
1012 DF900Z        753 GOTO   DF900   (1107)

                       ENTRY  DF400
     DF400                                   PROGRAM MODE
1013                 1 GOSUB  INIT5
1014                 0
1015              1670 C=REGN 14                 GET SS 1/2
```

```
641 1016            1474 RCR      1
642 1017            1530 ST=C
643 1020             514 ?S6=1                      DATAENTRY FLAG?
644 1021             213 GONC    DF410  (1042) NO
645 1022               1 GOSUB   GETPC              PRINT DATAENTRY STRING
645 1023               0
646 1024              14 ?S3=1                      ALPHAMODE?
647 1025               1 GSUBNC INCADA              NO. SKIP OVER NULL AT
647 1026               0
648                                                 BEGINNING OF DIGIT ENTRY STRING
649 1027               1 GOSUB   NXBYTA              FROM PROGRAM MEMORY
649 1030               0
650 1031             510 S6=      1                 SET UP FOR
651 1032            1610 S0=      1                  PPGS35
652 1033             212 B=A     WPT                MOVE ADDR TO B[3:0]
653 1034            1634 PT=      0                 SAVE FC
654 1035             130 G=C                         IN G FOR PPGS35
655 1036               1 GOSUB   PPGS35
655 1037               0
656 1040               1 GOSUB   EOLL
656 1041               0
657 1042 DF410       630 C=M                        PUT PTEMP2
658 1043            1530 ST=C                        TO ST
659 1044             114 ?S4=1                      "INSERT" BIT?
660 1045            1413 GONC    DF300  (1006) NON-PROGRAMMABLE FUNCTION
661 1046            1514 ?S12=1                     PRIVATE PGM?
662 1047             407 GOC     DF900  (1107) YES. DON'T PRINT ANYTHING.
663 1050               1 GOSUB   GETPC              A( 0-3 )= PC
663 1051               0
664 1052               1 GOSUB   SKPLIN             TEST FOR PC AT AN END
664 1053               0
665 1054               1 GOSUB   GETLIN             C( X )= LINE#, EN CHIP 0
665 1055               0
666 1056            1346 ? C#0   X                  LINE NUMBER= 000?
667 1057              33 GONC    DF414  (1062) YES
668 1060             514 ?S6=1                      NO, WAS IT AN END?
669 1061              27 GOC     DF415  (1063) YES
670 1062 DF414      1046 C=C+1   X                  INC LINE #
671                      LEGAL
672 1063 DF415         1 GOSUB   LINELB             LINE # TO PRINTER
672 1064               0
673 1065             630 C=M                        IS FC=ALBL OR LBLNN?
674 1066            1274 RCR      7
675 1067             126 C=0     XS                 FC TO
676 1070             406 A=C     X                   A.X
677 1071             460 LDI
678 1072             315 CON2     12       13       CD=ALBL
679 1073            1546 ? A#C   X                  FC#ALBL?
680 1074             353 GONC    DF420  (1131) ALBL
681 1075             460 LDI
682 1076             317 CON2     12       15       CF=LBL NN
683 1077            1546 ? A#C   X                  FC#LBL NN?
684 1100             313 GONC    DF420  (1131) LBL NN
685 1101               1 GOSUB   PBLANK
685 1102               0
686 1103 DF440         1 GOSUB   CPFKB
686 1104               0
687 1105               1 GOSUB   EOLL
687 1106               0
* FALL INTO DF900 HERE
```

```

                         ENTRY   DF905
1107 DF900         1 GOSUB   DATP30          CHECK ERROR FLAG
1110               0
* ON RETURN FROM PDAT30, S9 IS CLEAR
1111 DF905       630 C=M
1112            1376 ?C#0    S               RESTORE S9
1113              23 GONC    DF910  (1115)
1114            1110 S9=     1
1115 DF910      1634 PT=     0
1116             130 G=C                     RESTORE PTEMP2 TO G
1117            1074 RCR     2
1120             346 BC EX   X               RESTORE 3D ARG TO B.X
1121            1074 RCR     2
1122             134 PT=     4
1123             412 A=C     WPT             RESTORE FC TO A[4:1]
1124             274 RCR     5
1125             530 M=C                     RESTORE XADR TO M[3:0]
1126            1166 C=C-1   XS              INIT N.X FOR CLP
1127             160 N=C
1130            1740 RTN

1131 DF420         1 GOSUB   PRTMSG          LABEL - PUT IN A DIAMOND
1132               0
1133             400 CON     @400            DIAMOND
1134            1473 GOTO    DF440  (1103)
*
                         EJECT
```

```
*************************************************************************
************************   STKPLT   *************************************
*************************************************************************
   718                          ENTRY   STKPLT
   719 1135               224 CON     @224          T
   720 1136                17 CON     @17           O
   721 1137                14 CON     @14           L
   722 1140                20 CON     @20           P
   723 1141                13 CON     @13           K
   724 1142                24 CON     @24           T
   725 1143                23 CON     @23           S
   726 1144 STKPLT          1 GOSUB   IACHR
   726 1145                 0
   727 1146               110 S4=     1             S4=1 TO SHOW STKPLT
   728 1147               133 GOTO    RPLT00 (1162)
*************************************************************************
************************   REGPLT   *************************************
*************************************************************************
   732                          ENTRY   REGPLT
   733 1150               224 CON     @224          T
   734 1151                17 CON     @17           O
   735 1152                14 CON     @14           L
   736 1153                20 CON     @20           P
   737 1154                 7 CON     @7            G
   738 1155                 5 CON     @5            E
   739 1156                22 CON     @22           R
   740 1157 REGPLT          1 GOSUB   IACHR
   740 1160                 0
   741 1161               104 S4=     0             S4=0 TO SHOW REGPLT
   742 1162 RPLT00          1 GOSUB   GETVAL        REG A= MAX, REG M= MIN
   742 1163                 0
   743 1164               256 C=A                   COPY MAX TO C
   743 1165               416
   744 1166                 1 GOSUB   ACKC          ERROR IF MAX= ALPHA
   744 1167                 0
   745 1170               630 C=M                   MIN TO C
   746 1171                 1 GOSUB   ACKC          ERROR IF MIN= ALPHA
   746 1172                 0
   747 1173               630 C=M                   REG C= MIN
   748 1174                 1 GOSUB   A-C           MAX - MIN
   748 1175                 0
*IF (MAX-MIN) OVER/UNDER FLOWS THEN THE NUMBERS ARE TOO FAULTY TO BE ABLE
*TO PLOT, SO GIVE "DATA ERROR".
   751
   752 1176              1524 ?PT=    12            RESULTS OK?
   753 1177   -            23 GONC    RPLTDE (1201) NO, OVER/UNDER FLOW= "DATA ERROR"
   754 1200              1356 ? C#0                 MAX = MIN?
   755 1201 RPLTDE          1 GOLNC   ERRDE         YES, "DATA ERROR"
   755 1202                 2
   756 1203              1376 ? C#0   S             NO, MAX < MIN?
   757 1204              1757 GOC     RPLTDE (1201) YES, "DATA ERROR"
   758 1205               160 N=C                   N= MAX-MIN
   759 1206                 1 GOSUB   GETVAL        A= MAX
   759 1207                 0
   760 1210               316 C=B                   C= Y VALUE
   761 1211                 1 GOSUB   ACKC          ERROR IF Y VALUE= ALPHA
   761 1212                 0
   762 1213               316 C=B                   C= Y VALUE (SIGN DESTROYED BY ACKC)
   763 1214                 1 GOSUB   A-C           MAX - Y VALUE
```

```
1215                 0
*FOR (MAX-Y) AN UNDERFLOW IS OK AND PERFECTLY LEGITIMATE FOR "Y" VERY
*CLOSE TO "MAX".      JUST SET (Y-MIN) = (MAX-MIN) SINCE Y=MAX.
*AN OVERFLOW CAN OCCUR FOR 2 CASES:
*CASE 1 -- MAX<0 AND Y>0.   THIS MEANS Y>MAX SO IT WILL BE CAUGHT AND
*          Y WILL BE MADE EQUAL TO MAX.
*CASE 2 -- MAX>0 AND Y<0.   SINCE (MAX-MIN) DIDN'T OVERFLOW, Y WOULD HAVE
*          TO BE LESS THAN "MIN", WHICH WILL BE CAUGHT IN THE TEST OF
*          Y<MIN.

1216              1376 ? C#0   S              Y VALUE > MAX?
1217                33 GONC    Y<MIN? (1222) NO
1220               260 C=N                    YES, Y-MIN= MAX-MIN SINCE Y=MAX
1221               123 GOTO    RPLT20 (1233)
1222 Y<MIN?          1 GOSUB   GETVAL         B= Y VALUE, M= MIN
1223                 0
1224               156 AB EX                  A= Y VALUE
1225               630 C=M                    C= MIN
1226                 1 GOSUB   A-C            Y VALUE - MIN
1227                 0
*FOR (Y-MIN) AN UNDERFLOW IS OK AND PERFECTLY LEGITIMATE FOR Y VERY CLOSE
*TO MIN.     JUST SET (Y-MIN)=0.
*AN OVERFLOW CAN OCCUR IN 2 CASES:
*CASE 1 -- Y<0 AND MIN>0.  THIS MEANS Y<MIN WHICH IS HANDLED BY MAKING
*          Y-MIN=0 WHICH IS THE SAME AS SETTING Y=MIN.
*CASE 2 -- Y>0 AND MIN<0.  SINCE THIS POINT IN THE CODE IS ONLY REACHED
*          WHEN Y<=MAX, AND MAX-MIN DIDN'T OVERFLOW, THIS CASE IS IMPOSSIBLE.

1230              1376 ? C#0   S              Y VALUE < MIN?
1231                23 GONC    RPLT20 (1233) NO
1232               116 C=0                    YES, SET Y VALUE-MIN= 0
1233 RPLT20       1150 REGN=C  9              REG 9= Y VALUE-MIN
1234                 1 GOSUB   GETVAL         C= NNN.AAA
1235                 0
1236               530 M=C                    SAVE COPY OF NNN.AAA
1237                 1 GOSUB   ACKC           ERROR IF NNN.AAA= ALPHA
1240                 0
1241               630 C=M                    RESTORE C= NNN.AAA
1242              1004 S2=     0
1243              1376 ? C#0   S              NNN.AAA < 0?
1244                33 GONC    GETNNN (1247) NO
1245              1010 S2=     1              YES
1246               136 C=0     S              MAKE IT POSITIVE
1247 GETNNN        210 S5=     1              GET INTEGER PART
1250              1240 SETDEC
1251                 1 GOSUB   INTFRC         GET  NNN
1252                 0
1253              1356 ? C#0                  NNN= 0?
1254              1253 GONC    RPLTDE (1201) YES, "DATA ERROR"
1255               416 A=C                    A= NNN
1256               116 C=0
1257              1534 PT=     12
1260               120 LC      1              C= 1
1261                 1 GOSUB   A-C            C= NNN - 1
1262                 0
*NNN IS A POSITIVE INTEGER AT THIS POINT SO OVER/UNDER FLOW IS NOT POSSIBLE
*BY SUBTRACTING A "1".

1263              1140 SETHEX
1264               530 M=C                    SAVE NNN-1 IN FLOATING FORM
```

```
 817 1265           1 GOSUB  CONV3C         CONVERT NNN-1 TO BINARY
 817 1266           0
 818 1267         406 A=C    X              A= NNN-1
 819 1270         460 LDI
 820 1271         250 CON    168
 821 1272        1406 ? A<C  X              NNN-1 < 168?
 822 1273 RPLTER 1063 GONC   RPLTDE (1201) NO, "DATA ERROR"
 823 1274        1270 C=REGN 10             YES
 824 1275         246 AC EX  X              C= NNN-1  (BINARY)
 825 1276        1250 REGN=C 10             STORE NNN-1 IN REG 10
 826 1277         630 C=M                   RESTORE F.P. VALUE OF NNN-1
 827 1300         416 A=C                   A= NNN-1    (F.P.)
 828 1301         260 C=N                   C= MAX - MIN    (F.P.)
 829 1302        1240 SETDEC
 830 1303           1 GOSUB  DV2-10         (NNN-1)/(MAX-MIN)
 830 1304           0
(MAX-MIN) AND (NNN-1) ARE KNOWN TO BE VALID NUMBERS.
SINCE 0<= (NNN-1)< 168  UNDERFLOW IS HARD TO GET AND RESULTS IN VVV=0 OR
 (AAA-1)=0 WHICH IS OK SO DON'T CHECK, BUT AN OVERFLOW COULD
HAPPEN FOR VERY SMALL (MAX-MIN).
 835
 836 1305           1 GOSUB  OVFL10         CHECK OVERFLOW
 836 1306           0
 837 1307         324 ?PT=   10             OVERFLOW?
 838 1310        1637 GOC    RPLTER (1273) YES, "DATA ERROR"
 839 1311         160 N=C                   N= (NNN-1)/(MAX-MIN)
 840 1312         416 A=C
 841 1313        1170 C=REGN 9              C= Y - MIN
 842 1314           1 GOSUB  INTCAL         C=INT[(Y-MIN)(NNN-1)/(MAX-MIN) + 0.5
 842 1315           0
 843 1316         406 A=C    X              A= VVV
 844 1317         460 LDI
 845 1320           3 CON    3
 846 1321        1106 C=A-C  X              C= VVV-3
 847 1322          23 GONC   RPLT30 (1324) VVV<3?
 848 1323         106 C=0    X              YES, VVV-3= 0
 849 1324 RPLT30  674 RCR    11             VVV-3 TO C(3-4)
 850 1325         416 A=C
 851 1326        1270 C=REGN 10
 852 1327         406 A=C    X              NNN-1 TO A(X)
 853 1330         134 PT=    4
 854 1331         252 AC EX  WPT            VVV-3, NNN-1 TO "C"
 855 1332        1250 REGN=C 10             R10(X)=NNN-1,R10(3-4)=VVV-3
 856 1333        1014 ?S2=1                 SUPPRESS AXIS?
 857 1334          43 GONC   RPLT40 (1340) NO
 858 1335          74 RCR    3              YES, SET AAA-1 = VVV-3
 859 1336         126 C=0    XS
 860 1337         523 GOTO   RPLT50 (1411)
 861 1340 RPLT40    1 GOSUB  GETVAL         C= NNN.AAA
 861 1341           0
 862 1342        1240 SETDEC
 863 1343         204 S5=    0              GET FRACTIONAL PART
 864 1344           1 GOSUB  INTFRC         GET .AAA
 864 1345           0
 865 1346        1346 ? C#0  X              .AAA=0?
 866 1347         257 GOC    RPLT45 (1374) NO
 867 1350           1 GOSUB  GETVAL         YES, A= MAX, M= MIN
 867 1351           0
 868 1352        1516 ? A#0                 MAX=0?
 869 1353          33 GONC   AAA005 (1356) YES
 870 1354          1536 ? A#0   S                NO, MAX < 0?
 871 1355            33 GONC    AAA010 (1360) NO
 872 1356 AAA005 1270 C=REGN  10               YES, AAA-1= NNN-1
 873 1357           323 GOTO    RPLT50 (1411)
 874 1360 AAA010  630 C=M                      C= MIN
 875 1361          1376 ? C#0   S                MIN => 0?
 876 1362            37 GOC     AAA015 (1365) NO
 877 1363           116 C=0                     YES, AAA-1= 0
 878 1364           253 GOTO    RPLT50 (1411)
 879 1365 AAA015 1240 SETDEC
 880 1366          1276 C=-C-1 S                CHANGE (MIN) TO (-MIN)
 881 1367           416 A=C                     A= -MIN
 882 1370           260 C=N                     C= (NNN-1)/(MAX-MIN)
 883 1371             1 GOSUB   INTCAL          C=INT[-MIN(NNN-1)/(MAX-MIN)+0.5]
 883 1372             0
 884 1373           163 GOTO    RPLT50 (1411)
 885 1374 RPLT45  406 A=C     X                A= EXP OF .AAA
 886 1375           460 LDI
 887 1376             3 CON     3
 888 1377          1006 C=A+C   X                MULTIPLY .AAA BY 1000
 889 1400          1140 SETHEX
 890 1401             1 GOSUB   CONV3C          CONVERT TO BINARY
 890 1402             0
 891 1403           406 A=C     X                A= AAA
 892 1404           646 A=A-1   X                A= AAA-1
 893 1405          1270 C=REGN  10               C(0-1)= NNN-1
 894 1406           246 AC EX   X                A=NNN-1, C= AAA-1
 895 1407          1406 ? A<C   X                NNN-1 < AAA-1?
 896 1410          1467 GOC     AAA005 (1356) PEG AXIS AT RIGHT MARGIN
 897 1411 RPLT50  204 S5=     0
 898 1412          1150 REGN=C  9                R9(X)= AAA-1
 899 1413           674 RCR     11
 900 1414           432 A=C     M                A(M)= AAA-1
 901 1415          1270 C=REGN  10               C= NNN-1
 902 1416           406 A=C     X                A= NNN-1
 903 1417           460 LDI
 904 1420             6 CON     6
 905 1421           706 A=A-C   X                A= NNN-7
 906 1422            23 GONC    RPLT52 (1424) NNN < 7?
 907 1423             6 A=0     X                YES
 908 1424 RPLT52  206 B=A     X                B= NNN-7
 909 1425            74 RCR     3                C= VVV-3
 910 1426           126 C=0     XS
 911 1427          1616 A SR
 912 1430          1616 A SR
 913 1431          1616 A SR
 914 1432          1406 ? A<C   X                AAA-1 < VVV-3?
 915 1433           423 GONC    RPLT56 (1475) NO
 916 1434           530 M=C                     M= VVV-3
 917 1435          1446 ? A<B   X                AAA-1 < NNN-7?
 918 1436            47 GOC     RPLT75 (1442) YES, PLOT AXIS LINE
 919 1437           306 C=B     X                NO, C= NNN-7= SKIP
 920 1440            46 B=0     X                #RCOL= 0
 921 1441           433 GOTO    RPLT61 (1504) SKIP COLUMNS & PLOT VALUE
 922 1442 RPLT75    1 GOSUB   SKPC4           SKPCOL= A(X)= AAA-1
 922 1443             0
 923 1444             1 GOSUB   INITSC          SEND OUT MODE= SPECIAL CHAR
 923 1445             0
 924 1446             1 GOSUB   PRTMSG
 924 1447             0
```

```
925 1450            567 CON     @567            AXIS LINE
926 1451            146 A=B     X               A= NNN-7
926 1452            206
927 1453            630 C=M                     C=VVV-3
928 1454           1406 ? A<C   X               NNN-7 < VVV-3?
929 1455             27 GOC     RPLT80 (1457) YES
930 1456            406 A=C     X               NO, A=VVV-3
931 1457 RPLT80    1170 C=REGN 9                C= AAA-1
932 1460           1056 C=C+1                   C= (AAA-1)+1= AAA
933 1461            706 A=A-C   X               A= "A" - AAA= SKIP
934 1462            146 AB EX   X               B=SKIP, A=NNN-7
935 1463            706 A=A-C   X               A= NNN-AAA-7
936 1464            306 C=B     X               C= SKIP
937 1465            153 GOTO    RPLT60 (1502)
*
939 1466 SPLT90     404 S8=     0               NORMAL MODE
940 1467              1 GOSUB   INITSM          SEND MODE
940 1470              0
941 1471              1 GOSUB   PRTMSG
941 1472              0
942 1473            401 CON     @401            LITTLE X
943 1474            373 GOTO    RPLT65 (1533)
944
945 1475 RPLT56     146 AB EX   X               NO,A= NNN-7, B= AAA-1
946 1476           1406 ? A<C   X               NNN-7 < VVV-3?
947 1477             33 GONC    RPLT60 (1502) NO, C=VVV-3
948 1500            246 C=A     X               YES, C= NNN-7
948 1501            406
949 1502 RPLT60     706 A=A-C   X               A= # REMAINING COLUMNS
950 1503            206 B=A     X               B= #RCOL
951 1504 RPLT61       1 GOSUB   SKPCOM          SKIP TO CHARACTER
951 1505              0
952 1506 RPLT62     114 ?S4=1                   STKPLT?
953 1507           1577 GOC     SPLT90 (1466) YES
954 1510           1570 C=REGN 13               NO, REGPLT
955 1511             74 RCR     3               GET USER REG 0 POINTER
956 1512            406 A=C     X               A= R0 PTR
957 1513            460 LDI
958 1514              3 CON     3
959 1515           1006 C=A+C   X               C= R3 PTR
960 1516           1160 DADD=C
961 1517             70 C=DATA                  GET USER REG 3= SPECIAL CHAR
962 1520           1176 C=C-1   S
963 1521           1176 C=C-1   S               ALPHA DATA?
964 1522           1443 GONC    SPLT90 (1466) NO, USE DEFAULT CHAR
965 1523            416 A=C                     SAVE SPEC CHAR
966 1524              1 GOSUB   INITSC          SEND OUT MODE= SPECIAL CHAR
966 1525              0
967 1526           1334 PT=     13
968 1527            620 LC      6
969 1530            256 AC EX                   A(S)=6 FOR ACSPCC, C= SPEC CHAR
970 1531              1 GOSUB   ACSPCC          SEND OUT SPECIAL CHAR
970 1532              0
971 1533 RPLT65    1270 C=REGN 10               GET VVV-3
972 1534             74 RCR     3
973 1535            126 C=0     XS
974 1536            406 A=C     X               A= VVV-3
975 1537            460 LDI
976 1540              7 CON     7
977 1541            506 A=A+C   X               A= VVV + 4
```

```
1542           1170 C=REGN 9                  C=AAA-1
1543            246 AC EX  X                  A= AAA-1, C= VVV+4
1544           1406 ? A<C  X                  AAA-1 < VVV+4?
1545            157 GOC    RPLT70 (1562) YES
1546           1106 C=A-C  X                  NO, C= AAA-VVV-5= SKIP
1547            146 AB EX  X                  A= #RCOL
1550            706 A=A-C  X                  A= NEW #RCOL= #RCOL-SKIP
1551            646 A=A-1  X                  SUBTRACT 1 COL FOR AXIS
1552            206 B=A    X                  B= NEW #RCOL
1553              1 GOSUB  SKPCOM             SKPCOL
1554              0
1555              1 GOSUB  INITSC             SEND OUT MODE= SPEC CHAR
1556              0
1557              1 GOSUB  PRTMSG
1560              0
1561            567 CON    @567               AXIS LINE
1562 RPLT70     306 C=B    X                  C= # REMAINING COLUMNS
1563              1 GOSUB  SKPCOM             SKPCOL
1564              0
1565            404 S8=    0
1566              1 GOSUB  INITSM              GET OUT OF COLUMN MODE
1567              0

                    ENTRY  RPECHK
1570 RPECHK       1 GOSUB  EOLR               SEND RIGHT END OF LINE
1571              0
1572              1 GOLONG PECHK              CHECK FOR ERRORS
1573              2
                    EJECT
```

```
*
1574 GTSTK     70 C=DATA
1575          356 BC EX                      B= Y VALUE
1576          170 C=REGN 1
1577          530 M=C                        M= Y MIN
1600          270 C=REGN 2
1601          416 A=C                        A= Y MAX
1602          370 C=REGN 3                   C= NNN.AAA
1603         1740 RTN
*
*********************************************************************
*-GETVAL= GET VALUES
*
*-GETS Y MIN, Y MAX, NNN.AAA FROM USER REGS 0-3 FOR REGPLT, OR FROM
* STK X-Z FOR STKPLT.
*-ALSO GETS Y VALUE FROM X FOR REGPLT, OR FROM T FOR STKPLT
*
*-USES:    A,B,C,M,    NO PT,      S4,         NO SUB LEVELS
*-INPUTS:  S4=1 FOR STKPLT,     S4=0 FOR REGPLT
*-OUTPUTS: A= Y MAX,    B= Y VALUE,      C= NNN.AAA,    M= Y MIN,
*              CHIP 0 ENABLED,   HEXMODE
*
                          ENTRY  GETVAL
1604 GETVAL  106 C=0      X
1605        1160 DADD=C
1606        1140 SETHEX
1607         114 ?S4=1                       STKPLT?
1610        1647 GOC      GTSTK  (1574) YES
1611        1570 C=REGN 13                   NO
1612          74 RCR      3                  GET USER REG 0 POINTER
1613         416 A=C                         A= POINTER
1614        1160 DADD=C
1615          70 C=DATA                      GET Y MIN
1616         530 M=C                         M= Y MIN
1617         256 AC EX                       C= POINTER
1620        1056 C=C+1
1621         416 A=C
1622        1160 DADD=C
1623          70 C=DATA                      GET Y MAX
1624         256 AC EX                       A= Y MAX
1625        1056 C=C+1
1626        1160 DADD=C
1627          70 C=DATA                      GET NNN.AAA
1630         356 BC EX                       B= NNN.AAA
1631         116 C=0
1632        1160 DADD=C
1633         370 C=REGN 3                    C= VALUE
1634         356 BC EX                       C= NNN.AAA,   B= Y VALUE
1635        1740 RTN
*********************************************************************
*
* NPFTST - NON-PRINTING FCN TEST
* NON-PRINTING FUNCTIONS ARE:    PRA   A748
*                               PRBUF A74A
*                                ADV   8F
* RTNS TO P+1 IF FC IS ONE OF THE ABOVE
* RTNS TO P+2 IF FC IS NOT ONE OF THE ABOVE
* USES: C, A3:0, PT
* IN: M8:5=FC, LEFT JUSTIFIED
* OUT: NOTHING
* ASSUMES: NOTHING
*
                       ENTRY   NPFTST
1636 NPFTST   630 C=M
1637          274 RCR      5
1640           34 PT=      3              INPUT FC TO A3:0
1641          412 A=C      WPT
1642         1220 LC       10
1643          720 LC       7
1644          420 LC       4
1645         1220 LC       10             A74A=FC FOR PRBUF
1646           34 PT=      3
1647         1552 ? A#C    WPT            FC#PRBUF?
1650         1640 RTN NC
1651         1152 C=C-1    WPT
1652         1152 C=C-1    WPT            A748=FC FOR PRA
1653         1552 ? A#C    WPT            FC#PRA?
1654         1640 RTN NC
1655          112 C=0      WPT
1656         1020 LC       8
1657          143 GOTO     NPFTSC (1673)
*
*******************************************************************
* DON'T EVER CHANGE THE FOLLOWING "FILLTO @1637" !!!!!!!!!!!     *
*******************************************************************
                       FILLTO @1657
*
1660          205 CON      @205           E
1661           62 CON      @62            2
1662           40 CON      @40
1663           22 CON      @22            R
1664            5 CON      @05            E
1665           24 CON      @24            T
1666           16 CON      @16            N
1667           11 CON      @11            I
1670           22 CON      @22            R
1671           20 CON      @20            P
1672           55 CON      @55            -
     PHEAD             ENTRY   PHEAD    67BB
1673 NPFTSC  1720 LC       15             8F=FC FOR ADV
1674           34 PT=      3
1675         1552 ? A#C    WPT            FC#ADV?
1676         1640 RTN NC
1677            1 GOLONG RTNP+2
1700            2
*
*******************************************************************
* FMT - FORMAT FUNCTION                                           *
*******************************************************************
*
                       ENTRY   FMT
1701          224 CON      @224           T
1702           15 CON      @15            M
1703            6 CON      @06            F
1704 FMT      460 LDI
1705          300 CON      @300           SEND FORMAT COMMAND
1706          406 A=C      X
1707            1 GOLONG ACCHRX
1710            2
```

```
  1117
*
***********************************************************************
  1120                       ENTRY  BPROMT
  1121                       ENTRY  BPROM
  1122                       ENTRY  BPROM1
  1123 1711 BPROMT  246 AC EX  X             FC TO C
  1124 1712 BPROM1    1 GOSUB  PPROM1        SEND FC PROMPT TO PRINTER
  1124 1713           0
  1125 1714 BPROM  1076 C=C+1  S             COUNT THE BLANK
*
*BPROM FALLS INTO PBLANK HERE.
*
***********************************************************************
  1130
* EOLR - SEND AN EOLR USING CPBYTE
  1132
* EOLL - SEND AN EOLL USING CPBYTE
  1134
* THE PIL PRINTER WILL NOT USE EOLR OR EOLL AS A DELIMINATOR ANY MORE,
* INSTEAD EOLR & EOLL WILL BE USED AS PRINT MODE CONTROLL.
* BOTH EOLR & EOLL WILL CHECK WHAT IS LAST EOL, IF NOT THE SAME WE
* WANT TO SEND THIS TIME, WILL SEND AN EOLR OR EOLL AND THEN SEND
* CR&LF.
* PBLANK - SEND A BLANK USING CPBYTE
*
***ALL USE: C(X),N,  NO PT, NO STS,  NO ADDITIONAL SUB LEVELS
***PRINT IF FLAG 55=1,  DON'T PRINT IF FLAG 55=0  (FLAG 55= PRINTER EXISTAN
  1144
  1145                       ENTRY  PBLANK
  1146 1715 PBLANK  460 LDI
  1147 1716          40 CON    @40           BLANK
  1148 1717         353 GOTO   EOLR10 (1754)
  1149                       ENTRY  EOLR
  1150                       ENTRY  EOLCR
  1151 1720 EOLR    644 C=HPIL 6             GET LAST STATUS 2ND BYTE
  1151 1721         672
  1151 1722         603
  1152 1723        1474 RCR    1
  1153 1724         776 C=C+C  S             TEOL = 1 ?
  1154 1725         137 GOC    EOLCR  (1740) YES, LAST EOL WAS A EOR
  1155 1726         460 LDI
  1156 1727         350 CON    @350          EOLR
  1157 1730 EOLMCH  144 HPL=CH 1             WRITE DATA CONTROL BITS
  1158 1731           5 CH=    @001
  1159 1732        1200 HPIL=C 2             SEND EOLR OR EOLL
  1160 1733 EOLM10  354 ORAV?                READY FOR NEXT FRAME ?
  1161 1734          47 GOC    EOLCR  (1740) YES
  1162 1735        1046 C=C+1  X             TIME OUT ?
  1163 1736        1753 GONC   EOLM10 (1733) NOT YET
  1164 1737 EOLER  1740 RTN
7E0 1165 1740 EOLCR   144 HPL=CH 1
  1166 1741           5 CH=    @001
  1167 1742         244 HPL=CH 2
  1168 1743          65 CH=    @15           SEND "CR"
  1169 1744         106 C=0    X
  1170 1745 WATCR   354 ORAV?                CR COMES BACK YET ?
  1171 1746          47 GOC    EOL    (1752) YES, SEND "LF"
  1172 1747        1046 C=C+1  X             TIME OUT YET ?
  1173 1750        1753 GONC   WATCR  (1745) NOT YET
  1174 1751           1663 GOTO   EOLER  (1737) TRANSMIT ERROR
  1175 1752 EOL        460 LDI     .
  1176 1753            12 CON    @12            LOAD "LF"
  1177 1754 EOLR10      1 GOLONG CPBYTE         SEND IT
  1177 1755             2
  1178
  1179                     ENTRY  EOLL
67EE1180 1756 EOLL     644 C=HPIL 6
  1180 1757            672
  1180 1760            603
  1181 1761           1166 C=C-1  XS
  1182 1762           1046 C=C+1  X              TALKING TO T.V. ?
  1183 1763           1557 GOC    EOLCR  (1740) YES, SUPRESS EOLL
  1184 1764           1146 C=C-1  X
  1185 1765           1474 RCR    1
  1186 1766            776 C=C+C  S              TEOL = 0 ?
  1187 1767           1513 GONC   EOLCR  (1740) YES, LAST EOL WAS AN EOLL
  1188 1770            460 LDI
  1189 1771            340 CON    @340           EOLL
  1190 1772           1363 GOTO   EOLMCH (1730)
*
*****************************************************************
*
* NXBTXP - GET NEXT BYTE, USING S6 TO DECIDE ROM/RAM
* USES: C, A3:0, AND 1 ADDITIONAL SUBROUTINE LEVEL
* IN: A3:0=ADDRESS
*     S6=1  FOR ROM, S6=0 FOR RAM
*     PT=3
* OUT: A3:0 INCREMENTED TO NEXT BYTE ADDRESS
*     C1:0=NEXT BYTE
* ASSUMES: HEXMODE, ANY DATA STORAGE CHIP ENABLED
*
  1203                     ENTRY  NXBTXP
  1204 1773 NXBTXP   514 ?S6=1                 ROM?
  1205 1774            1 GOLNC  NXBYTA         NO
  1205 1775            2
  1206 1776            1 GOLONG NXBYTO         YES
  1206 1777            2
  1207
*
  1209                     UNLIST
  1212                     END

  ERRORS :       0
```

```
SYMBOL TABLE

AAA005   1356    -   1410 1353
AAAC10   1360    -   1355
AAAC15   1365    -   1362
ACREGC    316    -    314
ACXSUB    315    -
ALPD45    340    -    342
ALPD50    343    -    351
ALPD55    352    -    337
ALPDAT    327    -    321
BPROM    1714    -
BPROMI   1712    -
BPROMT   1711    -
DATA&F    622    -
DATA&P    616    -
DATAPR     41    -
DATP15     66    -     62
DATP17     72    -     65
DATP20     75    -     67
DATP25     77    -     40
DATP30    104    -     74
DF05J     711    -    646
DF10      643    -    641
DF15      653    -    650
DF20      713    -    676
DF200    1001    -    714
DF300    1006    -   1045  775   772
DF40      724    -    716
DF400    1013    -
DF410    1042    -   1021
DF414    1062    -   1057
DF415    1063    -   1061
DF420    1131    -   1100 1074
DF440    1103    -   1134
DF50      740    -    723
DF60      773    -    767
DF70      776    -    744
DF900    1107    -   1047 1012
DF900T    731    -    700
DF900X    709    -    652
DF900Y   1003    -
DF900Z   1012    -   1003 1000   731
DF905    1111    -
DF910    1115    -   1113
EOL      1752    -   1746
EOLCR    1740    -   1767 1763 1734 1725
EOLER    1737    -   1751
EOLL     1756    -
EOLM10   1733    -   1736
EOLMCH   1730    -   1772
EOLR     1720    -
EOLR10   1754    -   1717
FMT      1704    -
GETNNN   1247    -   1244
GETVAL   1604    -
GTSTK    1574    -   1610
HFFISC   1673    -   1657
```

```
NPFTST  1636   -
NXBTXP  1773   -
NXIN10   460   -    456
NXIN15   501   -    473   470
NXIN21   504   -    476
NXIN30   530   -    521
NXIN70   546   -    560
NXIN75   556   -    553
NXIN80   561   -    555   551
NXIN90   574   -    567
NXIN99   614   -    612
NXINST   450   -
OUTRG9   264   -    261
OVERFL    32   -
PBLANK  1715   -
PDIG10   365   -    363
PDIG25   400   -    446
PDIG30   417   -    403
PDIG48   442   -    437
PDIG50   444   -    432
PDIGAB   360   -    326
PDIGAC   356   -    264
PDIGE    113   -
PDIGX5   415   -    411
PHEAD   1673   -
PRQUOT   352   -
PRTDEF   115   -
PRTM     313   -
PRTMS1     1   -      6
PRTMSG     0   -     31    24    16
PRTMSL    17   -
PVIEW    265   -
PVW10    311   -    274
REGPLT  1157   -
RG9P10   132   -    136
RG9P13   135   -    131
RG9P17   143   -    145
RG9P19   144   -    142
RG9P20   154   -    152   141
RG9P24   160   -    162
RG9P26   165   -    202
RG9P27   166   -    173
RG9P28   174   -    167
RG9P29   203   -    155
RG9P30   205   -    171   164
RG9P32   220   -    224
RG9P33   223   -    217
RG9P34   226   -    222
RG9P35   227   -    211   207   204
RG9P40   255   -    251
RG9P42   257   -    245
RG9P45   260   -    256   254
RG9P50   262   -    241
RPECHK  1570   -
RPLT00  1162   -   1147
RPLT20  1233   -   1231  1221
RPLT30  1324   -   1322
RPLT40  1340   -   1334
RPLT45  1374   -   1347
RPLT50  1411   -   1373  1364  1357  1337
```

```
RPLT52  1424   -   1422
RPLT56  1475   -   1433
RPLT60  1502   -   1477 1465
RPLT61  1504   -   1441
RPLT62  1506   -
RPLT65  1533   -   1474
RPLT70  1562   -   1545
RPLT75  1442   -   1436
RPLT80  1457   -   1455
RPLTDE  1201   -   1273 1254  1204  1177
RPLTER  1273   -   1310
SPLT90  1466   -   1522 1507
STKPLT  1144   -
WATCR   1745   -   1750
Y<MIN?  1222   -   1217
```

```
ENTRY TABLE

ACREGC    316    -
ACXSUB    315    -
BPROM    1714    -
BPROMI   1712    -
BPROMT   1711    -
DATA&F    622    -
DATA&P    616    -
DATAPR     41    -
DATP25     77    -
DATP30    104    -
DF400    1013    -
DF905    1111    -
EOLCR    1740    -
EOLL     1756    -
EOLR     1720    -
FMT      1704    -
GETVAL   1604    -
HPFTST   1636    -
NXBTXP   1773    -
NXINST    450    -
OVERFL     32    -
PBLANK   1715    -
PDIGAB    360    -
PDIGAC    356    -
PDIGE     113    -
PHEAD    1673    -
PRQUOT    352    -
PRTDEF    115    -
PRTM      313    -
PRTMSG      0    -
PRTMSL     17    -
PVIEW     265    -
REGPLT   1157    -
RPECHK   1570    -
STKPLT   1144    -
```

```
EXTERNAL  REFERENCES

A-C       1174   1214   1226   1261
A-C       1175   1215   1227   1262
ACCHRX    1707
ACCHRX    1710
ACKC      1166   1171   1211   1237
ACKC      1167   1172   1212   1240
ACREGC     302
ACREGC     303
ACSPCC    1531
ACSPCC    1532
ACXSUB      36
ACXSUB      37
CKANGL     343
CKANGL     344
CKEN       267
CKEN       270
CKTRCE     466
CKTRCE     467
CLR&SS     524    570
CLR&SS     525    571
CONV3C    1265   1401
CONV3C    1266   1402
CPBYTE       2     27    354   1754
CPBYTE       3     30    355   1755
CPFKB      750   1006   1103
CPFKB      751   1007   1104
DATP25     703
DATP25     704
DATP30    1107
DATP30    1110
DF400      660
DF400      661
DF905      711
DF905      712
DV2-10    1303
DV2-10    1304
EOLL        72   1040   1105
EOLL        73   1041   1106
EOLR       102    564   1010   1570
EOLR       103    565   1011   1571
ERRDE     1201
ERRDE     1202
ERRPR      526
ERRPR      527
FILLIN     773
FILLIN     774
FILLNP     776
FILLNP     777
FLINKA     514
FLINKA     515
FNDPTR     272    471    644
FNDPTR     273    472    645
FNSTS      546
FNSTS      547
FORMAT     322
FORMAT     323
```

```
GETLIN  1054
GETLIN  1055
GETPC   1022  1050
GETPC   1023  1051
GETPCA   504
GETPCA   505
GETVAL  1162  1206  1222  1234  1340  1350
GETVAL  1163  1207  1223  1235  1341  1351
GLINE#   544
GLINE#   545
IACHR   1144  1157
IACHR   1145  1160
IAUNA     33    54
IAUNA     34    55
INCADA  1025
INCADA  1026
INIT5    113   732  1004  1013
INIT5    114   733  1005  1014
INITC    275   531
INITC    276   532
INITSC  1444  1524  1555
INITSC  1445  1525  1556
INITSM  1467  1566
INITSM  1470  1567
INTCAL  1314  1371
INTCAL  1315  1372
INTFRC  1251  1344
INTFRC  1252  1345
LBLCK    537
LBLCK    540
LDDP10   146   212
LDDP10   147   213
LINEL8  1063
LINEL8  1064
LOAD3    126
LOAD3    127
NFRPU    707
NFRPU    710
NOPRT    614
NOPRT    615
NPFTST   742  1001
NPFTST   743  1002
NXBYTA  1027  1774
NXBYTA  1030  1775
NXBYTO  1776
NXBYTO  1777
OVFL10  1305
OVFL10  1306
PAD1+A   770
PAD1+A   771
PAREG     70   734
PAREG     71   735
PBLANK  1101
PBLANK  1102
PBYTDU   345
PBYTDU   346
PBYTEC   370   375   425   442
PBYTEC   371   376   426   443
PDIGE    701   717
PDIGE    702   720
```

```
PECHK   1572
PECHK   1573
PEDIAG   111    572
PEDIAG   112    573
PPGMST    63
PPGMST    64
PPGS35  1036
PPGS35  1037
PPGSNL   562
PPGSNL   563
PPROM1  1712
PPROM1  1713
PRIORT   307
PRIORT   310
PRQUOT   330
PRQUOT   331
PRTDEF    75
PRTDEF    76
PRTMSG    77   1131   1446   1471   1557
PRTMSG   100   1132   1447   1472   1560
PUTPCD   512
PUTPCD   513
PUTPCF   602
PUTPCF   603
PWAIT     14
PWAIT     15
RPECHK   304
RPECHK   305
RSTSEQ   107    705
RSTSEQ   110    706
RTNP+2  1677
RTNP+2  1700
RUNING   464
RUNING   465
SKPC4   1442
SKPC4   1443
SKPCOM  1504   1553   1563
SKPCOM  1505   1554   1564
SKPLIN  1052
SKPLIN  1053
UNL      105    477    522    604
UNL      106    500    523    605

End of VASM assembly
*************************************************************
VASM ROM ASSEMBLY            REV.  6/81A

OPTIONS: L C S

    2                     FILE   SCPR3B
*
* ROW JUMP TABLE FOR PPGMST
*
    6     0            213 GOTO   PROW0  (   21)
    7     1            243 GOTO   PROW1  (   25)
    8     2            253 GOTO   PROW2  (   27)
    9     3            333 GOTO   PROW3  (   36)
   10     4            263 GOTO   PRW4-8 (   32)
   11     5            253 GOTO   PRW4-8 (   32)
   12     6            243 GOTO   PRW4-8 (   32)
```

```
  7            233 GOTO   PRW4-8 (   32)
 10            223 GOTO   PRW4-8 (   32)
 11            413 GOTO   PROW09 (   52)
 12            723 GOTO   PROW10 (  104)
 13            403 GOTO   PROW11 (   53)
 14            423 GOTO   PROW12 (   56)
 15            473 GOTO   PR1314 (   64)
 16            463 GOTO   PR1314 (   64)
 17              1 GOLONG PTXROW
 20              2
 21 PROW0      460 LDI
 22            317 CON2   12       15     PROMPT STRING IN C,F
 23 PRW010     646 A=A-1  X               OPERAND MINUS ONE
                   LEGAL
 24            143 GOTO   PPS120 (   40)
 25 PROW1        1 GOLONG PDEROW          THIS IS A DIGIT ENTRY ROW
 26              2
 27 PROW2      460 LDI
 30            220 CON2   9        0      PROMPT STRING IN 9,0
 31             73 GOTO   PPS120 (   40)
 32 PRW4-8       1 GOSUB  PPROMT
 33              0
 34              1 GOLONG OUTPPS
 35              2
 36 PROW3      460 LDI
 37            221 CON2   9        1      PROMPT STRING IN 9,1
 40 PPS120       2 A=0    PT              A( 1 ) _ 0
 41            206 B=A    X               SAVE THE OPERAND IN B
 42              1 GOSUB  PPROM1          OUTPUT PROMPT STRING
 43              0
 44              1 GOSUB  BPROM
 45              0
 46            436 A=C    S               A( S )= CHAR COUNTER
 47            306 C=B    X               C.X _ OPERAND
 50              1 GOLONG PRW930
 51              2
 52 PROW09     663 GOTO   PROW9  (  140)
 53 PROW11     460 LDI
 54            320 CON2   13       0      PROMPT STRING IN 13,0
 55           1463 GOTO   PRW010 (   23)
 56 PROW12     460 LDI
 57            316 CON2   12       14
 60           1406 ? A<C  X               IS IT LBLNN ?OR X<>NN?
 61            643 GONC   PRW910 (  145)  YES
 62              1 GOLONG PRW120
 63              2
 64 PR1314    1634 PT=    0
 65              2 A=0    PT
 66            246 AC EX  X               PRINT "GTO " OR "XEQ "
 67              1 GOSUB  PPROM1
 70              0
 71              1 GOSUB  BPROM
 72              0
 73            376 BC EX  S               CHAR CTR TO B( S )
 74            156 AB EX                  A( 0-3 )= PC, A( S )= CHAR CTR
 75              1 GOSUB  INCAD           SKIP ONE BYTE( THREE BYTE FC )
 76              0
 77              1 GOSUB  NXTBYT          GET 3RD BYTE ( LBL )
100              0
101           1730 CST EX
```

```
 102           1204 S7=      0
 103            743 GOTO     PRW935 ( 177)
 104 PROW10     460 LDI
 105            250 CON2     10      8         TEST FOR XECROM FC
 106           1406 ? A<C    X                 IS IT AN XECROM FC ?
 107              1 GOLC     PXROM             YES
 110              3
 111            460 LDI                        NO
 112            256 CON2     10      14
 113           1406 ? A<C    X                 IS IT AN XEQ/GTO IND ?
 114            317 GOC      PRW910 ( 145) NO
**NOTE: FC (10,15) WILL BE PRINTED AS AN XEQ/GTO IND.
 115              1 GOSUB    NBYTAB            GET OPERAND
 116              0
                    ENTRY    PR1010            FOR CPFKB
 117 PR1010     346 BC EX    X                 OPERAND TO "B"
 120            460 LDI
 121            320 CON2     13      0         LOAD GTO FC
 122            406 A=C      X                 A= GTO FC
 123            306 C=B      X                 OPERAND TO "C"
 124           1434 PT=      1
 125            742 C=C+C    PT                IS IT AN XEQ?
 126             23 GONC     PR1020 ( 130) NO, A GTO
 127            542 A=A+1    PT                YES, "A"= XEQ FC
                    LEGAL
 130 PR1020       1 GOSUB    PPROMT            FC PROMPT TO PRINTER
 131              0
* SUBROUTINE LEVELS RESTRICTED TO 2 HERE FOR CPFKB
 132              1 GOSUB    BPROM
 133              0
 134            436 A=C      S                CHAR CTR TO A(S)
 135            306 C=B      X                OPERAND TO "C"
 136           1730 CST EX                    C= STATUS BITS, ST= OPERAND
 137            223 GOTO     PRW933 ( 161)
*
* NUMERICAL OPERAND
* ROW 9
*
 140 PROW9      510 S6=      1                S6= 1 GIVES 1 DIGIT OUTPUT
 141            460 LDI
 142            234 CON2     9       12        TEST FOR 1 OR 2 DIGIT OPERAND
 143           1406 ? A<C    X                1 DIGIT OPERAND ?
 144             23 GONC     PRW911 ( 146) YES
*
* NUMERICAL OPERAND
* B[3:0] HAS ADDR POINT TO ONE BYTE BEFORE OPERAND
* IF S0=1 MEANS 1 DIGIT OPERAND
* IF S0=0 MEANS 2 DIGITS OPERAND
*
 145 PRW910     504 S6=      0                SET FLAG FOR 2 DIGIT OPERAND
 146 PRW911     246 AC EX    X                PRINT  THE FUNCTION FIRST
 147              1 GOSUB    PPROM1
 150              0
 151              1 GOSUB    BPROM
 152              0
 153            376 BC EX    S                B(S)= CHAR CTR
 154              1 GOSUB    NBYTAB           AB EX, GET OPERAND
 155              0
* ENTRY PRW930 FOR CPFKB
* USES: A,B,C,PT,N + 2 SUBROUTINE LEVELS
* INPUT:                   A(S)= CHAR CTR, C(0-1)= OPERAND + + + + + + +
* OUTPUT: # CHARS IN C.M, CHIP 0 ENABLED
* ASSUMES: HEXMODE, PT=P
  117                       ENTRY   PRW930
  118   156 PRW930 1730 CST EX                    MOVE OPERAND TO STATUS BITS
  119   157        1214 ?S7=1                     INDIRECT ?
  120   160         173 GONC    PRW935 ( 177) NO
  121                       ENTRY   PRW933
  122   161 PRW933 1204 S7=     0                 YES, CLEAR IND BIT OF OPERAND
  123   162        1730 CST EX                    "C"= OPERAND.STATUS TO "ST"
  124   163         406 A=C     X                  OPERAND TO "A"
  125   164         504 S6=     0                 TWO DIGIT OPERAND
  126   165           1 GOSUB   PRTMSG            PRINT "IND "
  126   166           0
  127   167         111 CON     @111              I
  128   170         116 CON     @116              N
  129   171         104 CON     @104              D
  130   172         440 CON     @440                  BLANK
  131   173        1334 PT=     13
  132   174         420 LC      4                 COUNT 4 CHARS
  133   175         536 A=A+C   S
  134                       LEGAL
  135   176          33 GOTO    PRW936 ( 201)
  136   177 PRW935 1730 CST EX                    "C"= OPERAND, STATUS TO "ST"
  137   200         406 A=C     X                 A(1-0)  OPERAND
  138   201 PRW936   26 A=0     XS
  139   202         460 LDI
  140   203         146 CON     102
  141   204        1406 ? A<C   X                 NUMERICAL OPERAND ?
  142   205         213 GONC    PRW940 ( 226) NO
  143   206         276 AC EX   S                 YES, CHAR CTR TO C(S)
  144   207          36 A=0     S
  145   210         576 A=A+1   S
  146   211         514 ?S6=1                     1 DIGIT NUMERICAL OPERAND ?
  147   212          27 GOC     PRW938 ( 214) YES, LEAVE A(S)= 1
  148   213         576 A=A+1   S                 NO,SET A(S)=2 TO GET 2 DIGITS
  149                       ENTRY   PRW938            FOR CPFKB
  150   214 PRW938 1036 C=C+A   S                 COUNT THE OPERAND CHARS
  151   215        1374 RCR     13                CHAR COUNT TO B(0)
  152   216         346 BC EX   X
  153   217           1 GOSUB   BINBCD
  153   220           0
* RESTRICTED TO 2 SUB LEVELS HERE FOR CPFKB
  155   221           1 GOSUB   PNUMBB            PRINT OPERAND
  155   222           0
  156   223         306 C=B     X
  157   224        1474 RCR     1                 CHAR CTR TO C(S)
  158   225         323 GOTO    OUTPPS ( 257)
*                 + + + + +  A(S)= CHAR CTR,    A(X)= OPERAND
  160                       ENTRY   PRW940
  161   226 PRW940  460 LDI
  162   227         164 CON     116
  163   230        1546 ? A#C   X                 IS IT A LSTX ?
  164   231         413 GONC    PL      ( 272) YES
  165   232        1406 ? A<C   X                 NO, IS IT A SMALL A-E?
  166   233         343 GONC    SMABC   ( 267) YES
  167   234         460 LDI
  168   235         160 CON     112
  169   236        1406 ? A<C   X                 CAPITAL A-J?
  170   237         257 GOC     CPABC   ( 264) YES
```

```
240          1546 ? A#C   X                  IS IT A T?
241           343 GONC    PT       ( 275) YES
*                      NO, IT IS Z,Y OR X
242          1046 C=C+1   X                  C(X)= 113
243           706 A=A-C   X                  A(X)= OFFSET
244           460 LDI
245           132 CON     @132               Z
246 PRW945    646 A=A-1   X
247            47 GOC     PRW960 ( 253)
250          1146 C=C-1   X
                  LEGAL
251          1753 GOTO    PRW945 ( 246)
252 PRW950   1106 C=A-C   X
253 PRW960    576 A=A+1   S                  COUNT THE CHAR
                  LEGAL
254             1 GOSUB   CPBYTE             SEND TO PRINTER
255             0
                  ENTRY   PPS200             FOR CPFKB
                  ENTRY   OUTPPS
256 PPS200    276 AC EX   S                  # CHARS TO "C"
257 OUTPPS    106 C=0     X
260           132 C=0     M
261           374 RCR     10                 # CHARS TO C(M)
262          1160 DADD=C                     ENABLE CHIP 0
263          1740 RTN
264 CPABC     460 LDI
265            45 CON     @45                LOAD OFFSET
266          1643 GOTO    PRW950 ( 252)
267 SMABC     460 LDI
270            32 CON     @32                LOAD OFFSET
                  LEGAL
271          1613 GOTO    PRW950 ( 252)
272 PL        460 LDI
273           114 CON     @114               L
274          1573 GOTO    PRW960 ( 253)
275 PT        460 LDI
276           124 CON     @124               T
277          1543 GOTO    PRW960 ( 253)
*
*
* ROW 1 - INCLUDING DIGIT ENTRY AND AGTO, AXEQ
* A[2:0] HAS THE FUNCTION CODE. B[3:0] POINTING 1ST BYTE OF
* DIGIT ENTRY STRING, IF ITS A DIGIT ENTRY FC.
*
                  ENTRY   PDEROW
300 PDEROW    460 LDI
301            35 CON2    1      13
302          1406 ? A<C   X                  IS IT A DIGIT ENTRY FC ?
303           603 GONC    PR0110 ( 363) NO, EITHER AGTO OR AXEQ
304            32 A=0     M                  YES, CLEAR CHAR COUNTER
305 PDER00    460 LDI
306            32 CON2    1      10
307          1406 ? A<C   X                  IS IT A DIGIT ?
310           267 GOC     PDER50 ( 336) YES
311          1546 ? A#C   X                  NO, IS IT A D.P.?
312           107 GOC     PDER10 ( 322) NO
313           460 LDI
314            56 CON     @56                ASCII D.P.
315           214 ?S5=1                      D.P. FLAG SET?
316           237 GOC     PDER55 ( 341) YES, SHOW D.P.
```

```
317        1146 C=C-1   X             C(X)= @54= ASCII COMMA
320        1146 C=C-1   X             C(X)= @54= ASCII COMMA
                LEGAL
321         203 GOTO    PDER55 ( 341)
322 PDER10 1046 C=C+1   X
323        1546 ? A#C   X             IS IT AN EEX ?
324          77 GOC     PDER20 ( 333) NO
325           1 GOSUB   PBLANK        YES, BLANK TO PRINTER
326           0
327         572 A=A+1   M             COUNT THE BLANK
330         460 LDI
331         105 CON     @105          E
332          73 GOTO    PDER55 ( 341)
333 PDER20  460 LDI                   IT MUST BE A CHS
334          55 CON     @55
335          43 GOTO    PDER55 ( 341)
336 PDER50  246 AC EX   X
337        1434 PT=     1
340         320 LC      3
341 PDER55  572 A=A+1   M             COUNT THE CHAR
                LEGAL
342           1 GOSUB   CPBYTE        SEND BYTE TO PRINTER
343           0
344           1 GOSUB   NBYTAB        AB EX, GET NEXT BYTE
345           0
346         156 AB EX                 B= PGM PTR, A(M)= CHAR COUNTER
347         126 C=0     XS
350         406 A=C     X             A.X _ NEXT BYTE
351         460 LDI
352          35 CON2    1       13
353        1434 PT=     1
354        1542 ? A#C   PT            IS THIS BYTE A ROW 1 FC ?
355          37 GOC     PDER90 ( 360) NO
356        1406 ? A<C   X             IS IT A DIGIT ENTRY FC ?
357        1267 GOC     PDER00 ( 305) YES
360 PDER90  272 AC EX   M             # CHAR CTR TO C(M)
361           1 GOLONG  ENCP00        ENABLE CHIP 0
362           2
                ENTRY   PR0110
*
** THE FC FOR "ASN" WILL NOT BE HANDLED VERY WELL!!!!!!!!!!!!
363 PR0110 1746 A SL    X             CONVERT FC FROM 1D TO D0
364          26 A=0     XS                       OR FROM 1E TO E0
365         246 AC EX   X             PRINT "GTO " OR "XEQ "
366           1 GOSUB   PPROM1
367           0
370           1 GOSUB   BPROM
371           0
372           1 GOSUB   CPYS6M
373           0
374           1 GOSUB   NXBTXP
375           0
376         173 GOTO    PSTRNG ( 415)

****************************************************************
*
* PSTRNG - PRINT TEXT STRING
* USES: C, A.S,A3:0, B.S, N, S9, AND 2 ADDITIONAL SUBROUTINE LEVELS
* IN: A3:0 = ADDRESS OF BYTE BEFORE FIRST CHARACTER
*     S6=1 IF ROM ADDRESS, S6=0 IF RAM ADDRESS
*     PT=3
*     C.0=LENGTH OF STRING
*     A.S=INCOMING CHAR COUNT
*     NOTE C.0+A.S MUST BE <= 15
* OUT: C.M=TOTAL CHAR COUNT (=C.0+A.S+2)
* ASSUMES: HEXMODE, S9=PRINTER INTERFACE ERROR FLAG

*
* PLBL - PRINT ALPHA LABEL
* USES: C, A.S,A3:0, B.S, N, S9, AND 2 ADDITIONAL SUBROUTINE LEVELS
* IN: A3:0 = ADDRESS OF 1ST BYTE OF LABEL
*     S6=1 FOR ROM, S6=0 FOR RAM
*     A.S = INCOMING CHARACTER COUNT (MUST BE <= 8)
* OUT: C.M=FINAL CHAR COUNT.
* ASSUMES: HEXMODE, S9=PRINTER INTERFACE ERROR FLAG

*
* PLBL0 - PRINT ALPHA LABEL WITH ZERO INCOMING CHAR COUNT
* ZEROES OUT A.S AND DROPS INTO PLBL

*
* PLBL3 - PRINT ALPHA LABEL WITH ADDR OF 3RD BYTE
* SAME AS PLBL EXCEPT FOR DIFFERENT INPUT.
* IN: A3:0=ADDRESS OF 3RD BYTE OF LABEL
*     S6=1 FOR ROM, S6=0 FOR RAM
*     A.S = INCOMING CHARACTER COUNT (MUST BE <= 8)
*     C.0 = LENGTH OF ALPHA LABEL, NOT COUNTING KEYCODE
*     PT=3

*
* PTXROW -- PRINT TEXT ROW
* SAME AS PSTRNG EXCEPT USES MORE & TAKES DIFFERENT INPUT
* USES: C, A.S,A3:0, B.S,B3:0, N, S9, & 1 ADDITIONAL SUB LEVEL
* IN: A3:0=ADDRESS OF BYTE BEFORE FIRST CHARACTER
*     S10=1 FOR ROM, S10=0 FOR RAM
*     A.0=LENGTH OF STRING
*
                        ENTRY  PTXROW
                        ENTRY  PSTRNG
                        ENTRY  PLBL
                        ENTRY  PLBL0
                        ENTRY  PLBL3
  377 PLBL0      36 A=0    S            INITIALIZE CHAR COUNT
  400 PLBL       34 PT=    3
  401             1 GOSUB  INADXP       INC ADDR
  402             0
  403             1 GOSUB  NXBTXP       GET 3RD BYTE
  404             0
  405 PLBL3       1 GOSUB  INADXP       POINT TO KEYCODE
  406             0
  407          1146 C=C-1  X            DEC LENGTH FOR KEYCODE
                    LEGAL
  410            53 GOTO   PSTRNG ( 415)

  411 PTXROW    246 AC EX  X            STRING LENGTH TO C.0
  412           136 C=0    S            INITIALIZE CHAR COUNT
  413             1 GOSUB  CPYS6M
  414             0

  415 PSTRNG   1474 RCR    1            STRING LENGTH TO C.S
```

```
416          276 AC EX   S                A.S=STRING LENGTH
                                          C.S=CHAR COUNT
417         1036 C=A+C   S
420          376 BC EX   S                SAVE TOTAL CHAR COUNT IN B.S
421          460 LDI
422           42 CON     @42              QUOTES
423 PSTR10     1 GOSUB   CKANGL
424            0
425            1 GOSUB   CPBYTE
426            0
427           34 PT=     3
430          676 A=A-1   S                DONE?
431           47 GOC     PSTR20 ( 435) YES
432            1 GOSUB   NXBTXP           GET NEXT BYTE
433            0
434         1673 GOTO    PSTR10 ( 423)

435 PSTR20     1 GOSUB   PRQUOT           PUT OUT QUOTE
436            0
437          116 C=0
440          336 C=B     S
441          374 RCR     10               TOTAL CHAR COUNT TO C.M
442         1072 C=C+1   M
443         1072 C=C+1   M                ADD 2 FOR QUOTES
444         1740 RTN
                 ENTRY   PRW120
**.....FUNCTION CODE= ALPHA LBL OR END ..................
445 PRW120   156 AB EX                    PGM PTR TO "A"
446          216 B=A                      & KEEP A COPY IN B
447            1 GOSUB   INCAD            SKIP LINK BYTE
450            0
451            1 GOSUB   NXTBYT           LOAD 3RD BYTE
452            0
453         1434 PT=     1
454         1042 C=C+1   PT               IS IT LBL ?
455          123 GONC    PRW122 ( 467) NO, ITS AN END
456          460 LDI                      FC=  LBL
457          317 CON2    12      15       LOAD LBL FC
460            1 GOSUB   PPROM1           PRINT  THE FUNCTION
461            0
462            1 GOSUB   BPROM
463            0
464            1 GOSUB   CPYS6M
465            0
466         1123 GOTO    PLBL   ( 400)
*
** ......FUNCTION CODE= END ..............
467 PRW122 1730 CST EX                    SET THE STATUS
470          314 ?S10=1                   ARE WE IN ROM ?
471          177 GOC     PRW124 ( 510) YES, PROMPT "END" ONLY
472          214 ?S5=1                    FINAL END ?
473          153 GONC    PRW124 ( 510) NO
474         1730 CST EX                   YES, RESTORE STATUS
                 ENTRY   PR.END           FOR PRINTING THE CATALOG
    PR.END
475            1 GOSUB   PRTMSG           PRINT ".END."
476            0
477           56 CON     @56              .
500          105 CON     @105             E
501          116 CON     @116             N
```

```
  388  502             104 CON    @104         D
  389  503             456 CON    @456         .
  390  504             116 C=0
  391  505              34 PT=     3
  392  506             520 LC      5           # CHAR CTR= 5
  393  507            1740 RTN
  394  510 PRW124 1730 CST EX                  RESTORE STATUS BITS
  395  511             460 LDI
  396  512             300 CON2    12      0    PRINT "END"
  397  513               1 GOSUB   PPROM1
  397  514               0
  398  515               1 GOLONG OUTPPS
  398  516               2
***********************************************************************
*-PPGMST= PRINT PROGRAM STEP
*-SENDS LINE# AND PROGRAM STEP TO PRINTER
*
*-PPGSNL= PRINT PROGRAM STEP, NO LINE NUMBER
*-SAME AS PPGMST EXCEPT ONLY SENDS LINE NUMBERS FOR LABELS
*
*-USES:     A,B,C,G,N,         PT,         S0-S7,           3 SUB LEVELS
*-INPUTS:    PC= LAST BYTE OF LAST INSTR,   REG F= VALID LINE #
*      S7=1 FOR PGM LISTING IF IN "ALL" (TRACE), ELSE S7= DON'T CARE
*-OUTPUTS: # OF CHARS IN C.M, CHIP 0 ENABLED
*
*
* PPGS35 - ENTRY POINT USED BY PRT5 IN PROGRAM MODE TO PRINT DATAENTRY
* STRINGS ONLY.
*
* USES A,B,C,G,N,PT,S0-S7
*
* INPUT: SET S6 (LINE# FLAG) AND S0 ("ADD BLANK" FLAG)
*      ADDR OF FIRST BYTE OF DATA ENTRY STRING IN MM FORM IN B[3:0]
*      FIRST BYTE OF DATA ENTRY STRING IN G
* OUTPUT: ONE LINE TO PRINTER
* ASSUMES: HEXMODE & PT=P.
*
  423                       ENTRY   PPGMRS
  424                       ENTRY   PPGSNL
  425                       ENTRY   PPGMST
  426                       ENTRY   PPGS35
  427  517 PPGSNL  504 S6=     0             CLEAR LINE # FLAG
  428  520          33 GOTO    PPGS05 ( 523)
  429  521 PPGMRS 1530 ST=C                  RESTORE STATUS
  430  522 PPGMST  510 S6=     1             SET LINE# FLAG
  431  523 PPGS05    1 GOSUB   GETPC         GET PROGRAM POINTER
  431  524           0
  432  525 PPGS10    1 GOSUB   NXTBYT        GET 1ST BYTE OF PROGRAM STEP
  432  526           0
  433  527        1434 PT=     1
  434  530        1352 ? C#0   WPT           NULL?
  435  531        1743 GONC    PPGS10 ( 525) YES, SKIP IT
  436  532        1610 S0=     1             NO, INITIALIZE "ADD BLANK" FLAG
  437  533           1 GOSUB   LBLCK         CHECK FOR LBL
  437  534           0
  438  535         114 ?S4=1                 FC= LBL?
  439  536         313 GONC    PPGS35 ( 567) NO
  440  537           1 GOSUB   FNSTS         YES, GET PRINTER STATUS
  440  540           0
  441  541          14 ?S3=1                 OOPS?
```

```
542            23 GONC    PPGS20 ( 544) NO
543          1110 S9=     1             SET ERROR FLAG
544 PPGS20    114 ?S4=1                 "ALL" MODE ?US
545            53 GONC    PPGS25 ( 552) NO
546          1730 CST EX                RESTORE STATUS
547          1214 ?S7=1                  PRINTING PROGRAM?
550            77 GOC     PPGS32 ( 557) YES
551           123 GOTO    PPGS33 ( 563) NO,BLANK LINE BEFORE LBL
552 PPGS25    214 ?S5=1                 NORM?
553            37 GOC     PPGS30 ( 556) YES
554          1730 CST EX
555           113 GOTO    PPGS34 ( 566)
556 PPGS30   1730 CST EX
557 PPGS32    776 C=C+C   S             LAST LINE.HAD EOLL?
560             1 GSUBNC  EOLL          NO, ADD EOLL
561             0
562           414 ?S8=1                 LAST LINE= LBL??
563 PPGS33      1 GSUBNC  EOLCR         NO, ADD BLANK LINE
564             0
565           510 S6=     1             SET LINE # FLAG
566 PPGS34   1604 S0=     0             CLEAR "ADD BLANK" FLAG
567 PPGS35    106 C=0     X
570          1160 DADD=C                ENABLE CHIP 0
571           204 S5=     0             CLEAR D.P. FLAG
572          1670 C=REGN  14            GET STATUS REG
573           534 PT=     6
574           742 C=C+C   PT            D.P. FLAG SET?
575            23 GONC    PPGS37 ( 577) NO
576           210 S5=     1             YES, SET D.P. FLAG
577 PPGS37    514 ?S6=1                 PRINT LINE# ?
600           153 GONC    PPGS65 ( 615) NO
601          1770 C=REGN  15            GET LINE #
602             1 GOSUB   BINBD0        LINE #: BIN TO BCD
603             0
604             1 GOSUB   LINELC        LINE # TO PRINTER
605             0
606           460 LDI
607            40 CON     @40           BLANK
610          1614 ?S0=1                 ADD A BLANK?
611            27 GOC     PPGS60 ( 613) YES
612           106 C=0     X             NO,  000= DIAMOND
613 PPGS60      1 GOSUB   CPBYTE        SEND DIAMOND TO PRINTER
614             0
615 PPGS65   1634 PT=     0
616           230 C=G                   GET SAVED FC
617           406 A=C     X             COPY OF FC IN "C" AND "A"
620            26 A=0     XS
621          1434 PT=     1             SET UP PT FOR JUMP TABLE
622           504 S6=     0             SET UP 2 DIGIT OPERAND FLAG
623          1074 RCR     2             SAVE FC
624           460 LDI                   GET ADDR OF JUMP TABLE
625          1500 CON     @1500
626           746 C=C+C   X             ADDR= @64000= 6800 HEX
627           374 RCR     10            FC ROW= LAST ADDR DIGIT
630           740 GOTOC                 TO ROW JUMP TABLE (@64000)
                  EJECT
```

```
***************************************************************************
* PPROMT= PRINT A PROMPT STRING FOR A MICROCODE FUNCTION
*
* PPROMT ENTRY: A[1:0]=MAINFRAME FC,     LEAVES PT= 2
* PPROM1 ENTRY: C[1:0]=MAINFRAME FC,     LEAVES PT= 2
* PPROM2 ENTRY: C[6:3]=XADR
*
* ALL ENTRY POINTS USE: A,C,N, NO PT, S0,S5,S9 FOR ERRORS,1 SUB LEVEL
*
*-INPUT:  A(0-1)= MAINFRAME FC
*-OUTPUT: C(S)= # CHARS
*         A.M=XADR
*-ASSUMES: NO PUNCTUATION IN MAINFRAME FC PROMPTS
*
                      ENTRY  PPROMT
                      ENTRY  PPROM1
                      ENTRY  PPROM2
 631 PPROMT  246 AC EX  X             FC TO C(X)
 632 PPROM1 1074 RCR    2
 633         460 LDI                  MAIN FUNCTION TABLE
 634          24 CON    @24             START FROM @12000 (CN5)
 635        1174 RCR    9             LAST 2 ADDR DIGITS= FC
 636        1460 CXISA                LOAD XADR= XDEF
 637          34 PT=    3
 640         120 LC     1
 641         674 RCR    11            CHANGE XDEF TO XEQ ADDR
 642 PPROM2  204 S5=    0             INITIALIZE FINAL CHAR FLAG
 643         136 C=0    S             INITIALIZE CHAR COUNTER
 644         432 A=C    M             SAVE XADR IN A.M
 645 PRNT20 1604 S0=    0             INITIALIZE SPECIAL CHAR FLAG
 646        1172 C=C-1  M
 647        1460 CXISA                GET CHARACTER
 650        1076 C=C+1  S             COUNT THE CHAR
 651         126 C=0    XS            UPPER BITS USED BY MAINFRAME
 652        1730 CST EX
 653         514 ?S6=1                SPECIAL CHARACTER?
 654          33 GONC   PRNT30 ( 657) NO
 655        1056 C=C+1                YES, SET SPEC CHAR FLAG (S0)
 656         504 S6=    0             CLEAR SPECIAL CHAR BIT
 657 PRNT30 1214 ?S7=1                FINAL CHARACTER?
 660          53 GONC   PRNT40 ( 665) NO
 661        1204 S7=    0             YES, CLEAR FINAL CHAR BIT
 662        1730 CST EX
 663         210 S5=    1             SET FINAL CHAR FLAG
 664          23 GOTO   PRNT45 ( 666)
 665 PRNT40 1730 CST EX
 666 PRNT45  160 N=C                  CTR,ADDRESS TO "N"
 667         406 A=C    X             CHAR TO A.X
***DON'T HAVE TO CHECK FOR ILLEGAL CHARS IN MAINFRAME PROMPTS
 670           1 GOSUB  LCDASC        LCD FORMAT CHAR TO ASCII
 671           0
 672         260 C=N
 673         246 AC EX  X             CHAR TO C.X
 674           1 GOSUB  CKANGB        SEE IF THE SIGMA SIGN
 675           0
 676         206 B=A    X             RESTORE B.X
 677           1 GOSUB  CPBYTE        CHAR TO PRINTER
 700           0
 701         214 ?S5=1                FINAL CHARACTER?
```

```
702        1433 GONC   PRMT20 ( 645) NO, GET NEXT ONE
703        1740 RTN                  YES
                EJECT
```

```
********************************************************************************
* PXROM - PRINT EXTERNAL ROM FUNCTION PROMPT
*
*-FINDS THE EXECUTION ADDRESS IN ROM, THEN PRINTS:
*      - THE PROMPT=  MICROCODE
*      - THE ALPHA LBL=  USER LANGUAGE
*
*-USES:    A, B, C,    N,   PT,      S6,S8        2 SUB LEVELS
*-INPUTS:  A(0-1)= 1ST BYTE OF 2 BYTE FC
*          B(0-3)= PC POINTING TO 1ST BYTE OF FC
*                  P SELECTED
*-OUTPUTS: C.M=CHAR COUNT
*          IF FCN IS IN MICROCODE, THEN XADR IS RETURNED IN A.M
*-ASSUMES: HEXMODE, S9=PRINTER INTERFACE ERROR FLAG

*
* PPXROM - PRINT PROMPT, BUT NOT ARGUMENT, FOR AN XROM FUNCTION
* USES: A, B, C, PT, S8:0, N, AND 2 ADDITIONAL SUBROUTINE LEVELS
* IN: C2:0=XROM FC, RIGHT THREE DIGITS
* OUT: C.M=CHAR COUNT
*      IF FCN IS IN MICROCODE, THEN XADR IS PRESERVED IN A.M
*      S7:0 OUT = 1:0 IN
* ASSUMES: HEXMODE, S9=PRINTER INTERFACE ERROR FLAG

*
* PXR10 - SPECIAL ENTRY POINT FOR CPFKB, WHICH ENTERS WITH S8=1 SO
* THAT, IF THE FCN IS IN MICROCODE, PXROM WILL EXIT WITH A GOLONG
* TO PFK20 TO ALLOW CPFKB TO TACK ON THE ARGUMENT, IF THERE IS ONE.
* CPFKB CAN'T AFFORD TO CALL PPXROM WITH A GOSUB BECAUSE OF NOT
* ENOUGH SUBROUTINE LEVELS.
*
                       ENTRY  PXROM
                       ENTRY  PPXROM
                       ENTRY  PXR10
   704 PXROM     246 AC EX  X                1ST BYTE TO "C"
   705          1574 RCR    12
   706           160 N=C                     SAVE 1ST BYTE
   707             1 GOSUB  NBYTAB            GET THE SECOND BYTE
   710             0
   711           406 A=C    X                2ND BYTE TO "A"
   712          1630 C=ST
   713           760 CN EX                   N(0-1)=STATUS,C(2-3)=1ST BYTE
   714          1434 PT=    1
   715           252 AC EX  WPT              2ND BYTE TO C(0-1)
   716 PPXROM    404 S8=    0                AVOID EXIT TO PFK20
   717 PXR10       1 GOSUB  GTRMAD           FIND IT IN THE ROM (NO CHIP 0)
   720             0
   721           163 GOTO   PXR19  ( 737) ROM NOT PLUGGED IN
   722            14 ?S3=1                   XTYPE=1?
   723           157 GOC    PXR20  ( 740) YES
   724           260 C=N                     MICROCODE FCN
   725          1530 ST=C                    RESTORE SAVED STATUS
   726           256 AC EX                   XADR TO C3:0
   727           674 RCR    11               XADR TO C.M
   730             1 GOSUB  PPROM2
   731             0
   732           414 ?S8=1                   SPEC EXIT FOR CPFKB?
   733 OUTPPX      1 GOLNC  OUTPPS           NO
   734             2
```

```
 607  735              1 GOLONG PFK20          YES
 607  736              2
 608
 609  737 PXR19      16 A=0                   SAY NOT FOUND
 610  740 PXR20     260 C=N                   RESTORE
 611  741          1530 ST=C                   SAVED STATUS
 612  742             1 GOSUB  PRTMSG         SEND "XROM" TO PRINTER
 612  743             0
 613  744           130 CON    @130           X
 614  745           122 CON    @122           R
 615  746           117 CON    @117           O
 616  747           115 CON    @115           M
 617  750           440 CON    @440              BLANK
 618  751          1516 ? A#0                 WAS THE FCN FOUND?
 619  752            73 GONC   PXR30  ( 761 ) NO
 620  753          1334 PT=    13
 621  754           520 LC     5
 622  755           436 A=C    S              CHAR COUNT TO A.S
 623  756           510 S6=    1              SAY ROM
 624  757             1 GOLONG PLBL
 624  760             2
*
* ROM NOT PLUGGED IN, DISPLAY ROM ID & FC #
*
 628  761 PXR30     332 C=B    M              C(M)= ROM ID
 629  762            74 RCR    3              ROM ID TO C(X)
 630  763             1 GOSUB  PBINB0         SEND ROM ID TO PRINTER
 630  764             0
 631  765           460 LDI
 632  766            54 CON    @54            ASCII COMMA
 633  767             1 GOSUB  CPBYTE         SEND TO PRINTER
 633  770             0
 634  771           306 C=B    X              FUNCTION # TO C(X)
 635  772             1 GOSUB  PBINB0         FUNCTION # TO PRINTER
 635  773             0
 636  774          1334 PT=    13
 637  775          1220 LC     10             CHAR CTR= 10 CHARS
 638  776          1353 GOTO   OUTPPX ( 733)
***************************************************************************
 640                    EJECT
```

```
**********************************************************************************
********************   LIST -- LIST NNN LINES   ***********************************
**********************************************************************************
   644  777            224 CON     @224          T
   645 1000             23 CON     @23           S
   646 1001            411 CON     @411          I
   647 1002            414 CON     @414          L
   648                     ENTRY   LIST
   649 1003 LIST         0 NOP                   NOP= NON-PROGRAMMABLE
   650 1004           1770 C=REGN  15            GET LINE #
   651 1005            106 C=0     X
   652 1006           1146 C=C-1   X             SET LINE # = FFF
   653 1007           1750 REGN=C  15            STORE LINE# = FFF
   654 1010            246 AC EX   X             # LINES TO "C"
   655 1011            463 GOTO    LISTN  (1057)
**********************************************************************************
********************   PRP   --   PRINT PROGRAM   *********************************
**********************************************************************************
   659 1012            220 CON     @220          P
   660 1013             22 CON     @22           R
   661 1014            420 CON     @420          P
* UPPER BIT IS ARGUMENT TYPE
   663                     ENTRY   PRP
   664                     ENTRY   PRPINT
   665 1015 PRP          0 NOP                    NOP SHOWS NON-PROGRAMMABLE
   666 1016           1610 S0=     1              DON'T RETURN TO PRPB
   667 1017 PRPINT    1170 C=REGN  9              RETRIEVE THE NAME
   668 1020            530 M=C                    SAVE FOR ASRCH
   669 1021           1356 ? C#0                  LABEL PRESENT?
   670 1022            153 GONC    PRTP15 (1037) NO
   671 1023              1 GOSUB   ASRCH          YES, GO DO ALPHA SEARCH
   671 1024              0
   672 1025           1356 ? C#0                  SUCCESS?
   673 1026 PRPERR       1 GOLNC   ERRNE          ERROR= "NONEXISTANT"
   673 1027              2
   674 1030           1114 ?S9=1                  MICROCODE?
   675 1031           1757 GOC     PRPERR (1026) YES, CAN'T LIST IT
   676 1032            304 S10=    0              CLEAR ROM FLAG
   677 1033           1014 ?S2=1                  ROM?
   678 1034            133 GONC    PRTP18 (1047) NO, RAM
   679 1035            310 S10=    1              YES, SET ROM FLAG
   680 1036            113 GOTO    PRTP18 (1047)
*
   682 1037 PRTP15     314 ?S10=1                 ROM FLAG?
   683 1040             43 GONC    PRTP16 (1044) NO
   684 1041              1 GOSUB   GETPC          YES, GET PGM POINTER
   684 1042              0
   685 1043             63 GOTO    PRTP20 (1051)
   686 1044 PRTP16       1 GOSUB   FLINKP         IN RAM,FIND END OF PGM
   686 1045              0
   687 1046            474 RCR     8
   688 1047 PRTP18      34 PT=     3
   689 1050            412 A=C     WPT
   690 1051 PRTP20       1 GOSUB   CPGMHD         FIND THE TOP OF THE PROGRAM
   690 1052              0
   691 1053              1 GOSUB   PUTPCF         STORE NEW PC, SET LINE# = FFF
   691 1054              0
   692 1055            106 C=0     X              LOAD LARGE # OF LINES SO IT
   693 1056           1146 C=C-1   X              WON'T STOP UNTIL AN END
```

```
                      ENTRY   LISTNB
1057 LISTN   1614 ?S0=1                    RETURN TO PRPB ?
1060         1640 RTN NC                   YES
1061          610 S11=    1                NOT IN BARCODE MODE
1062          132 C=0     M                CLEAR CHAR COUNTER
1063          134 PT=     4
1064          120 LC      1
1065         1020 LC      8                LOAD CHAR CTR= 24
1066         1150 REGN=C  9                SAVE CTRS IN REG 9
1067            1 GOSUB   IPRT             INITIALIZE FOR EXPLICIT PRINT
1070            0
1071         1651 CON     @1651            GOSUB @57752 IN TIMER ROM TO
1072          574 CON     @574             PRINT THE CURRENT TIME
1073            1 GOSUB   GLINE#           CALC & STO LINE#,CK PRIVATE
1074            0
1075 LISTNB     1 GOSUB   EOLL             CLEAR BUFFER OF MODE BYTE
1076            0
1077          410 S8=     1                1 BLANK LINE BEFORE PACK LISTING
1100           33 GOTO    PRTP40 (1103)
1101 PRTP30  1056 C=C+1
1102         1750 REGN=C  15               UPDATE LINE #
1103 PRTP40  1170 C=REGN  9                GET # LINES COUNTER
1104         1146 C=C-1   X                DONE WITH NNN LINES?
1105          567 GOC     OUTPRP (1163) YES
1106         1150 REGN=C  9                NO, UPDATE # LINES CTR

                      ENTRY   PRTP50
1107 PRTP50     1 GOSUB   FNSTS            GET NEW PRINTER STATUS
1110            0
1111          346 BC EX   X                SAVE STATUS IN B(X)
1112           14 ?S3=1                    OOPS?
1113           23 GONC    PRTP55 (1115) NO
1114         1110 S9=     1                SET ERROR FLAG
1115 PRTP55   114 ?S4=1                    TRACE?
1116          567 GOC     PRTPAC (1174) YES, PRINT PACKED LISTING
**NOTE: SWITCHING FROM PRINTER "ALL" (TRACE) MODE TO NORM OR MAN CAN
*       LEAVE A PARTIAL LINE IN THE PRINTER BUFFER.
*
1117          776 C=C+C   S                LAST LINE HAD EOL?
1120            1 GSUBNC  EOLL             NO, SEND EOLL
1121            0
1122            1 GOSUB   PWAIT            WAIT FOR THE PRINTER
1123            0
1124          306 C=B     X                BRING ORIG ST BACK TO C.X
1125          214 ?S5=1                    NORM?
1126          213 GONC    PRTPL  (1147) NO,MAN, PRINT LEFT JUSTIFIED
1127            1 GOSUB   PPGMRS           RESTORE STS,PRT FUNCT WITH LINE#.
1130            0
1131          404 S8=     0
1132          114 ?S4=1                    JUST PRINTED LBL?
1133           23 GONC    PRTP60 (1135) NO
1134          410 S8=     1                YES
1135 PRTP60   460 LDI
1136            7 CON     7
1137          406 A=C     X
1140           74 RCR     3                CHAR COUNT TO C.X
1141          706 A=A-C   X                CHAR COUNT>7?
1142            1 GSUBNC  PAD1+A           NO. PAD WITH BLANKS
1143            0
1144            1 GOSUB   EOLR             PRINT LINE RIGHT JUSTIFIED
```

```
1145               0
1146              53 GOTO    PRTP80 (1153)
1147 PRTPL         1 GOSUB   PPGMRS           RESTORE STS,PRT FUNCT WITH LINE #
1150               0
1151               1 GOSUB   EOLL             PRINT LINE LEFT JUSTIFIED
1152               0

                     ENTRY   PRTP80
1153 PRTP80        1 GOSUB   GETPC            GET PROGRAM POINTER,EN CHIP 0
1154               0
1155               1 GOSUB   SKPLIN           MOVE PC TO NEXT LINE
1156               0
*                       * SKPLIN SETS S6= 1 FOR AN END
1157               1 GOSUB   PUTPCL           STORE PROGRAM POINTER, GET LINE#
1160               0
1161             514 ?S6=1                    HIT AN END?
1162            1173 GONC    PRTP30 (1101) NO, CONTINUE
                     ENTRY   OUTPRP
1163 OUTPRP        1 GOSUB   FNSTS            YES, GET PRINTER STATUS
1164               0
1165             114 ?S4=1                    PACKED LISTING?
1166               1 GSUBC   EOLL             YES, FINISH PACKED LISTING
1167               1
1170             614 ?S11=1                   RETURN TO PRPB ?
1171            1640 RTN NC                   YES
1172               1 GOLONG PRX10             CHECK FOR ERRORS, GOLONG NFRPU
1173               2
*
1174 PRTPAC      306 C=B     X                RESTORE ORIG STATUS
1175            1530 ST=C
1176            1670 C=REGN 14
1177            1156 C=C-1                    CLEAR PRINT FLAG
1200            1650 REGN=C 14
1201               1 GOSUB   PPGSNL           COUNT THE CHARS
1202               0
1203             432 A=C     M                SAVE CHAR CTR
1204             572 A=A+1   M                A= (#CHAR +2 BLANKS) - 1
1205            1670 C=REGN 14
1206            1056 C=C+1                    SET PRINT FLAG
1207            1650 REGN=C 14
1210            1170 C=REGN 9                 GET # REMAINING CHAR POSITIONS
1211            1204 S7=     0                CLEAR "JUST FIT" FLAG
1212            1432 ? A<C   M                FITS WITH 2 BLANKS?
1213              77 GOC     PRPA20 (1222) YES, A= (#CHAR + 2 BLANKS) - 1
1214             672 A=A-1   M                NO
1215             672 A=A-1   M                SCRAP 2 BLANKS [A= #CHAR - 1]
1216            1432 ? A<C   M                FITS W/O 2 BLANKS?
1217             273 GONC    PRPA50 (1246) NO
1220 PRPA15      132 C=0     M                YES,MAKE # POSITIONS LEFT= 0
1221              43 GOTO    PRPA40 (1225)
1222 PRPA20      572 A=A+1   M                A= #CHAR + 2 BLANKS
1223             272 AC EX   M                "A"= # POS LEFT,"C"= # CHARS
1224            1132 C=A-C   M                UPDATE CHAR COUNT
1225 PRPA40     1150 REGN=C 9                 STORE IT
1226            1210 S7=     1                SET PROGRAM LISTING FLAG
1227               1 GOSUB   PPGSNL           PROGRAM STEP TO PRINTER
1230               0
1231             404 S8=     0                CLEAR LBL FLAG
1232             114 ?S4=1                    JUST PRINTED A LBL?
1233             107 GOC     PRPA48 (1243) YES
```

```
1234            1170 C=REGN 9                GET # POSITIONS LEFT
1235            1372 ? C#0  M                LAST STEP JUST FIT?
1236              63 GONC   PRPA49 (1244) YES, NO BLANKS
1237               1 GOSUB  PRTMSG           NO, SEND 2 BLANKS
1240               0
1241             642 CON    @642             SKIP 2 CHARACTERS
1242 PRPA45 1113 GOTO   PRTP80 (1153)
1243 PRPA48  410 S8=    1                SET LBL FLAG
1244 PRPA49 1210 S7=    1                SET "JUST FIT" FLAG
1245              32 A=0    M                CLEAR CHARACTER COUNTER
1246 PRPA50    1 GOSUB  EOLL             PRINT LEFT JUSTIFIED
1247               0
1250            1170 C=REGN 9                GET COUNTERS
1251             132 C=0    M                CLEAR CHAR COUNTER
1252             134 PT=    4                LOAD NEW CHAR CTR= 24
1253             120 LC     1
1254            1020 LC     8
1255            1150 REGN=C 9                STORE IT
1256            1432 ? A<C  M                # CHARACTERS <= 24?
1257            1413 GONC   PRPA15 (1220) NO, PRINT ON OWN LINE
1260               1 GOSUB  PWAIT            WAIT FOR THE PRINTER
1261               0
1262            1214 ?S7=1                   LAST STEP JUST FIT?
1263            1577 GOC    PRPA45 (1242) YES, GET NEXT STEP
1264               1 GOLONG PRTP50           NO, IT DIDN'T FIT AT ALL
1265               2
* CPFKB - COUNT OR PRINT FCN FROM KEYBOARD ENTRY
*
* PRESERVES: M
* USES: PT, A, B, C, N, S7:0, & 2 ADDITIONAL SUBROUTINE LEVELS
*
* INPUT: M[8:5] = 1 OR 2 BYTE FC, LEFT JUSTIFIED
*     IF FC IS XROM OR MAINFRAME NON-PROGRAMMABLE, M[4:2] MAY CONTAIN
*         AN ARGUMENT
*     FLAG 55=1 IMPLIES COUNT AND PRINT.  FLAG55=0 IMPLIES COUNT ONLY.
* OUTPUT: C.M=NUMBER OF CHARACTERS IN FCN DESCRIPTION
* ASSUMES: STD ASSUMPTIONS (PTR=P, HEXMODE, CHIP 0 ENABLED)
*
                         ENTRY  CPFKB
1266 CPFKB   630 C=M                     RETRIEVE FC
1267            1274 RCR    7                1ST BYTE OF FC TO C[1:0]
1270             126 C=0    XS
1271             416 A=C                     FC TO A[2:12]
1272             460 LDI
1273             315 CON2   12       13      CD=FC FOR ALBL FROM PARSE
1274            1546 ? A#C  X                FC#ALBL?
1275             343 GONC   PFK12  (1331) ALBL
1276             460 LDI
1277             240 CON2   10       0       LOW END OF XROM FC RANGE
1300            1406 ? A<C  X                FC<XROM?
1301             127 GOC    PFK10  (1313) YES
1302             460 LDI
1303             250 CON2   10       8       1 PAST XROM RANGE
1304            1406 ? A<C  X                FC=XROM?
1305              63 GONC   PFK10  (1313) NO
1306             630 C=M                     XROM
1307             274 RCR    5                XROM TO C[3:0]
1310             410 S8=    1                SET UP FOR PXR10
1311               1 GOLONG PXR10
1312               2
```

```

1313 PFK10      246 C=A     X             CONSTRUCT XADR
1314            406
1315            674 RCR     11
1316            534 PT=     6
1317            120 LC      1
1320            420 LC      4
1321           1460 CXISA
1322             34 PT=     3
1323            120 LC      1
                    ENTRY   PFK11         USED BY PXROM
1324 PFK11      674 RCR     11            XADR TO C.M
1325           1172 C=C-1   M             CONSTRUCT XADR-1
1326           1460 CXISA
1327           1346 ? C#0   X
1330            777 GOC     PFK17  (1427)
                    ENTRY   PFK300
     PFK12
     PFK300                               C(XADR-1)=0...NO PROMPT STRING
                                          OR ALBL
* COULD BE ALBL, GTOL, AGTO, AXEQ, XEQ/GTO IND, OR R/S FROM PRT8
1331           1334 PT=     13
1332            460 LDI
1333              5 CON     5             FC FOR R/S
1334           1546 ? A#C   X             FC#R/S?
1335            117 GOC     PFK310 (1346) NOT R/S
1336              1 GOSUB   PRTMSG
1337              0
1340            122 CON     @122          R
1341            125 CON     @125          U
1342            516 CON     @516          N
1343            320 LC      3
1344 PFK305       1 GOLONG  OUTPPS
1345              2

1346 PFK310     460 LDI
1347              1 CON     1             GTOL
1350           1546 ? A#C   X             FC#GTOL?
1351            267 GOC     PFK320 (1377) NOT GTOL
1352              1 GOSUB   PRTMSG
1353              0
1354            107 CON     @107          G
1355            124 CON     @124          T
1356            117 CON     @117          O
1357             40 CON     @40
1360            456 CON     @456          .
1361            630 C=M                   RETRIEVE ARGUMENT
1362           1074 RCR     2
1363           1046 C=C+1   X             GTO..?
1364             67 GOC     PFK315 (1372) YES
1365            520 LC      5             NO, CHAR COUNT
1366            436 A=C     S
1367           1046 C=C+1   X             GTO.ALPHA?
1370            327 GOC     PFK337 (1422) YES
1371            753 GOTO    PFK45  (1466) 3D (OR 4D) ARGUMENT

1372 PFK315       1 GOSUB   PRTMSG        GTO..
1373              0
1374            456 CON     @456          .
1375            620 LC      6             CHAR COUNT
```

```
1376         1463 GOTO   PFK305 (1344)

1377 PFK320   460 LDI
1400          256 CON2   10      14      FC=AE=XEQ/GTO IND
1401         1546 ? A#C  X               FC#XEQ/GTO IND?
1402           57 GOC    PFK330 (1407)
1403          630 C=M                    XEQ/GTO IND
1404          274 RCR    5               INDIRECT 2D ARGUMENT
1405            1 GOLONG PR1010
1406            2

1407 PFK330   460 LDI
1410          315 CON2   12      13      CD=ALBL
1411         1546 ? A#C  X               FC#ALBL?
1412          117 GOC    PFK340 (1423)
1413          460 LDI                    ALBL
1414          317 CON2   12 ·     15     CF=FC FOR LBL NN
1415 PFK334     1 GOSUB  PPROM1
1416            0
1417            1 GOSUB  BPROM           SEND & COUNT BLANK
1420            0
1421          436 A=C    S               CHAR COUNT TO A.S
1422 PFK337   623 GOTO   PFK52  (1504)

1423 PFK340   246 AC EX  X               AXEQ OR AGTO
1424          136 C=0    S               CONSTRUCT FC FOR
1425         1374 RCR    13               XEQNN OR GTONN
1426         1673 GOTO   PFK334 (1415)
*
1427 PFK17   1072 C=C+1  M               CONSTRUCT XADR AGAIN
                  LEGAL
1430            1 GOSUB  PPROM2
1431            0
                  ENTRY  PFK20
1432 PFK20    436 A=C    S               CHAR COUNT TO A.S
1433          272 AC EX  M               C.M=XADR
1434         1172 C=C-1  M               C.M=XADR-1
1435         1460 CXISA                  GET OP1 TO C.XS
1436         1366 ? C#0  XS              OP1#0?
* FOR <EY TO PARSE OPERAND TYPES (OP1, OP2) SEE DRC'S LAB BOOK #8338
* P.25
1437            1 GOLNC  PPS200          EXIT
1440            2
1441            1 GOSUB  PBLANK          ADD A BLANK
1442            0
1443          576 A=A+1  S               INC CHAR COUNT
1444         1460 CXISA                  RESTORE OP1 TO C.XS
1445          766 C=C+C  XS
1446          766 C=C+C  XS
1447          766 C=C+C  XS              OP1 BIT 1 SET?
1450          103 GONC   PFK38  (1460) NO
1451          504 S6=    0               SAY 2D ARGUMENT
1452           23 GOTO   PFK35  (1454)
1453 PFK34    510 S6=    1               1 DIGIT ARGUMENT
1454 PFK35    630 C=M                    PUT ARG
1455          274 RCR    5                IN C[1:0]
1456            1 GOLONG PRW930
1457            2

1460 PFK38   1172 C=C-1  M               C.M=XADR-2
```

```
1461             1460 CXISA                      GET OP2
1462             1166 C=C-1   XS
1463              217 GOC     PFK52  (1504) ALPHA OPERAND
1464             1166 C=C-1   XS
1465              153 GONC    PFK50  (1502)
                      ENTRY   PFK45
        PFK45                                    3D ARGUMENT
1466              630 C=M                        PUT ARG
1467             1074 RCR     2                   TO C.X
1470             1334 PT=     13                 SET A.S=3 TO GET 3D
1471              320 LC      3                   FROM BINBCD
1472              256 AC EX                      AND PUT ARG IN A.X
1473              460 LDI
1474             1750 CON     1000
1475             1406 ? A<C   X                  ARG < 4 DIGITS?
1476               27 GOC     PFK47  (1500) YES
1477              576 A=A+1   S                  NO, OUTPUT 4 DIGITS
                      LEGAL
1500 PFK47          1 GOLONG PRW938
1501                2

1502 PFK50       1166 C=C-1   XS
1503             1503 GONC    PFK34  (1453)
                      ENTRY   PFK52
     PFK52                                       ALPHA OPERAND
1504                1 GOSUB   PRQUOT
1505                0
1506              116 C=0
1507              276 AC EX   S                  MOVE CHAR COUNT TO C.S
1510              374 RCR     10                  NOW TO C.M
1511              432 A=C     M                    AND BACK TO A.M
1512             1170 C=REGN 9                   GET STRING
1513             1434 PT=     1
1514 PFK55        572 A=A+1   M                  INC CHAR COUNT
1515             1352 ? C#0   WPT                IS THERE A CHARACTER LEFT?
1516              103 GONC    PFK57  (1526) NO
1517                1 GOSUB   CKANGL             CHECK ANGEL SIGN
1520                0
1521                1 GOSUB   CPBYTE
1522                0
1523              112 C=0     WPT                ZERO OUT THIS CHAR
1524             1074 RCR     2                  ROTATE NEXT CHAR INTO POS
1525             1673 GOTO    PFK55  (1514)

1526 PFK57          1 GOSUB   PRQUOT
1527                0
1530              572 A=A+1   M                  INC CHAR COUNT
1531              630 C=M
1532             1274 RCR     7
1533             1434 PT=     1
1534              412 A=C     WPT
1535              460 LDI
1536               17 CON     15                 FC FOR ASN
1537             1552 ? A#C   WPT                FC#ASN?
1540               33 GONC    PFK70  (1543)
1541 PFK59        272 AC EX   M                  CHAR COUNT TO C.M
1542             1740 RTN

1543 PFK70          1 GOSUB   PBLANK             ASN
1544                0
```

```
 1545             572 A=A+1   M              INC CHAR COUNT
 1546             630 C=M
 1547             274 RCR     5              KC TO C1:0
 1550            1146 C=C-1   X              GET RID OF OFFSET
 1551            1530 ST=C                   KC TO S7:0
 1552               1 GOSUB   PRKC           PRINT KEYCODE
 1553               0
 1554            1653 GOTO    PFK59  (1541)
*************************************************************************
*-PNUMBR= NUMBER TO PRINTER
*
*-SENDS DIGIT STRING IN A(M) TO PRINTER
*    -THE # OF DIGITS IS DETERMINED BY A(S)
*
*-USES:   A(3-13), B(S), C, N,   NO PT,  NO STS,    1 SUB LEVEL
*-INPUTS: A(M)= DIGIT STRING  (LEFT JUSTIFIED)
*         A(S)= # DIGITS TO SEND TO PRINTER
*         HEX MODE
*-OUTPUTS: HEX MODE, CHIP 0 ENABLED, (IF # DIGITS PRINTED#0)
*
*
* PNUMBB - SAME AS PNUMBR EXCEPT EXPECTS # OF DIGITS IN B.S INSTEAD OF
*      A.S
*
                        ENTRY   PNUMBB
                        ENTRY   PNUMBR
                        ENTRY   PBINB0
                        ENTRY   PBINBD
 1555 PBINB0  136 C=0     S              OUTPUT 2,3, OR 4 DIGITS
 1556 PBINBD    1 GOSUB   BINBDC         CONVERT TO BCD
 1557           0
 1560 PNUMBB  176 AB EX   S              # DIGITS TO A(S)
 1561 PNUMBR  272 AC EX   M              DIGITS TO C(M)
 1562        1374 RCR     13             LEFT JUSTIFY DIGITS IN "C"
 1563 BNBCD3  676 A=A-1   S              COUNT DIGITS, DONE?
 1564        1540 RTN C                  YES
 1565         460 LDI                    NO
 1566           3 CON     3              ADD UPPER 4 BITS
 1567        1374 RCR     13             GET NEXT DIGIT
 1570           1 GOSUB   CPBYTE         SEND TO PRINTER
 1571           0
 1572        1713 GOTO    BNBCD3 (1563)
*************************************************************************
*-LINELB= LINE # WITH LEADING BLANKS TO PRINTER
*
*-INPUTS:  C(X)= LINE # (BINARY), HEXMODE
*-USES:    A,B(S),C,N,  ACTIVE PT,  NO STS,  2 ADDITIONAL SUB LEVELS
*-OUTPUTS: HEX MODE, CHIP 0 ENABLED  (IF # DIGITS PRINTED # 0)
*
                        ENTRY   LINELB
 1573 LINELB    1 GOSUB   BINBD0         LINE#: BIN TO BCD
 1574           0
                        ENTRY   LINELC
 1575 LINELC 1334 PT=     13
 1576         320 LC      3
 1577         176 AB EX   S              A.S= # OUTPUT DIGITS
 1600        1436 ? A<C   S              ADD LEADING BLANK?
 1601           1 GSUBC   PBLANK         YES
 1602           1
 1603        1563 GOTO    PNUMBR (1561) LINE # TO PRINTER
```

```
*************************************************************************
*-GCHAR=  GET CHARACTER (FROM DISPLAY)
*-LCDASC= LCD TO ASCII
*
*-GCHAR GETS A CHARACTER FROM THE DISPLAY AND CONVERTS IT TO ASCII
*
*-USES:   A(X),C, NO PT,  S0 (SPECIAL CHAR),      NO ADDITIONAL SUB LEVELS
*-INPUTS: [GCHAR]: DISPLAY ENABLED, RAM DISABLED
*         [LCDASC]: A(0-1)= LCD FORMAT CHAR WITH NO PUNCTUATION
*         [LCDASC]: C(12-13) WILL BE PRESERVED (IT IS OUTPUT AS PUNCTUATION)
*-OUTPUTS: A(0-1)= ASCII CHARACTER, C(12-13)= PUNCTUATION (=0 IF NO PUNCT)
*
                      ENTRY  GCHAR
                      ENTRY  LCDASC
 1604 GCHAR   1604 S0=    0            FETCH LEFT CHAR FROM DISPLAY
 1605         1770 RABCL               SCRAP GARBAGE BITS
 1606          766 C=C+C  XS           SCRAP GARBAGE BITS
 1607          766 C=C+C  XS           SCRAP GARBAGE BITS
 1610          766 C=C+C  XS           SPECIAL CHARACTER?
 1611          766 C=C+C  XS
 1612           23 GONC   GCHR40 (1614) NO
 1613         1610 S0=    1            YES
 1614 GCHR40   406 A=C    X            CHAR TO "A"  (XS= 0)
 1615          460 LDI
 1616          100 CON    @100
 1617          706 A=A-C  X            ANY PUNCTUATION?
 1620           33 GONC   GCHR45 (1623) YES
 1621          106 C=0    X            NO
 1622          173 GOTO   GCHR50 (1641)
 1623 GCHR45   706 A=A-C  X            PERIOD?
 1624           77 GOC    GCHR47 (1633) YES
 1625          706 A=A-C  X            NO, COLON?
 1626          107 GOC    GCHR48 (1636) YES
 1627          460 LDI                 NO, MUST BE COMMA
 1630           54 CON    @54          ASCII COMMA
 1631         1074 RCR    2            SAVE PUNCTUATION
 1632          123 GOTO   LCDASC (1644)
 1633 GCHR47   460 LDI
 1634           56 CON    @56          ASCII PERIOD
 1635           33 GOTO   GCHR49 (1640)
 1636 GCHR48   460 LDI
 1637           72 CON    @72          ASCII COLON
 1640 GCHR49  1074 RCR    2            SAVE PUNCTUATION
 1641 GCHR50   460 LDI
 1642          100 CON    @100
 1643          506 A=A+C  X            RESTORE UPPER BITS
 1644 LCDASC  1614 ?S0=1               SPECIAL CHAR?
 1645          107 GOC    SPCASC (1655) YES
 1646          460 LDI                 NO
 1647           40 CON    @40
 1650         1406 ? A<C  X            CHAR < @40 ?
 1651         1640 RTN NC              NO
 1652 REGASC   746 C=C+C  X            YES, C(X)= @100
 1653          506 A=A+C  X            ASCII= CHAR + @100
 1654         1740 RTN
 1655 SPCASC    74 RCR    3            SAVE PUNCTUATION IN C(9-10)
 1656          246 AC EX  X            LCD CHAR TO "C"
 1657         1474 RCR    1            SAVE DIGIT 0 OF CHAR
 1660          460 LDI
 1661         1300 CON    @1300        LOAD ADDR= @26000
```

```
  1127 1662          374 RCR    10              ADDR DIGIT 0= LCD DIGIT 0
  1128 1663         1460 CXISA                  GET ASCII EQUIVALENT FROM
*                          *SPECIAL CHAR TABLE, CN11, @0
  1130 1664          406 A=C    X
  1131 1665         1740 RTN
*
*
  1134                   FILLTO @1670
       1666         0000 NOP
       1667         0000 NOP
       1670         0000 NOP
*
*
***********************************************************************
* [redacted] - PRINT WHAT'S IN THE DISPLAY   6BB9
* PRTLCD
* USES: A(X&S),B.X,C,S0,S9,N,ACTIVE PTR, AND +1 SUB LEVEL
*
* INPUT: CONTENTS OF THE LCD REGISTERS
* OUTPUT: ONE LINE TO THE PRINTER BUFFER (NO EOL), CHIP 0 ENABLED.
* ASSUMES: HEXMODE. DOESN'T CARE WHICH CHIP IS ENABLED.
*
* NOTE: THIS ENTRY POINT USED BY TIMER ROM TOO. SO DON'T USE ANY
*       ADDITIONAL CPU REGS
*
  1149 6BB9 PRTLCD       ENTRY  PRTLCD
  1150 1671 [redacted] 1334 PT=    13
  1151 1672         1320 LC     11              SET UP COUNTER
  1152 1673          436 A=C    S                IN A.S
  1153 1674            1 GOSUB  ENLCD
  1153 1675            0
  1154 1676 PLCD10     1 GOSUB  GCHAR
  1154 1677            0
  1155 1700          246 AC EX  X
  1156 1701            1 GOSUB  CKANGB
  1156 1702            0
  1157 1703          146 AB EX  X               RESTORE B.X
  1158 1704            1 GOSUB  PBYTDU          C(X) TO PRINTER
  1158 1705            0
  1159 1706         1434 PT=    1
  1160 1707         1574 RCR    12
  1161 1710         1352 ?C#0   WPT             PUNCTUATION?
  1162 1711            1 GSUBC  PBYTEC          YES
  1162 1712            1
  1163 1713          676 A=A-1  S               DONE?
  1164 1714         1623 GONC   PLCD10 (1676) NO
  1165 1715            1 GOLONG ENCP00
  1165 1716            2
***********************************************************************
*-LBLCK= LABEL CHECK
*
*-CHECKS FUNCTION CODE FOR LBL. RTNS WITH S4=1 FOR LBL, ELSE S4=0.
*
*-USES:   A,B,C, G,     PT,      S4,        2 SUB LEVELS
*-INPUTS: A(0-3)= PC,       C(0-1)= FC
*-OUTPUTS: S4=1 FOR LBL,  ELSE S4=0.
*          PT= 1,      CHIP 0 NOT NECESSARILY ENABLED
*          RETURNS FC IN G INSTEAD OF C[1:0]
*          RETURNS PC IN B[3:0] INSTEAD OF A[3:0]
*
```

```
                       ENTRY  LBLCK
1717 LBLCK   104 S4=    0                CLEAR "EOLL AFTER LBL" FLAG
1720         216 B=A                     SAVE PC
1721         126 C=0    XS
1722         406 A=C    X                FC TO "A"
1723        1634 PT=    0
1724         130 G=C                     SAVE FC IN "G"
1725        1434 PT=    1
1726        1502 ? A#0  PT               SHORT NUMERIC LBL?
1727         253 GONC   LBLCK9 (1754) YES
1730         460 LDI                     NO
1731         316 CON2   12      14
1732        1542 ? A#C  PT               ROW 12 FUNCTION?
1733        1540 RTN C                   NO
1734        1546 ? A#C  X                YES, "X<> NN" ?
1735        1640 RTN NC                  YES, SO SEND A BLANK
1736        1406 ? A<C  X                NO, ALPHA LBL OR END?
1737         153 GONC   LBLCK9 (1754) NO,LONG NUMERIC LBL
1740          34 PT=    3                YES
1741         152 A=B    WPT              COPY PC TO "A"
1742         212
1743           1 GOSUB  INCAD            SKIP 2ND BYTE
1744           0
1745           1 GOSUB  INCAD            MOVE TO THIRD BYTE
1746           0
1747           1 GOSUB  GTBYT            GET 3RD BYTE
1750           0
1751        1434 PT=    1
1752        1042 C=C+1  PT               ALPHA LBL?
1753        1640 RTN NC                  NO, IT'S AN END
1754 LBLCK9  110 S4=    1                SET LBL FLAG
1755        1740 RTN


*****************************************************************
******** PRT3 -- BEGIN TO KEY IN ALPHA OPERAND  *****************
*****************************************************************
                       ENTRY  ALPHOP
1756 ALPHOP 1634 PT=    0
1757         230 C=G
1760         530 M=C                     SAVE G REG
1761           1 GOSUB  DATAPR           PRINT DIGIT ENTRY STRING
1762           0
1763         630 C=M
1764        1634 PT=    0
1765         130 G=C                     RESTORE G REG
1766           1 GOLONG PR3RT
1767           2
*
******************************************************************
*
* CPYS6M - COPY S10 TO S6 & MISCELLANEOUS OTHER STUFF
* USES: A.S,A3:0, B3:0, PT, S6
* IN: C.S=CHAR COUNT
*      B3:0=ADDRESS
*      S10=1 FOR ROM, S10=0 FOR RAM
* OUT: A.S=CHAR COUNT
*      A3:0=ADDRESS
*      S6=1 FOR ROM, S6=0 FOR RAM
*      PT = 3
* ASSUMES: NOTHING
*
  1234                        ENTRY   CPYS6M
  1235 1770 CPYS6M  436 A=C     S
  1236 1771          34 PT=     3
  1237 1772         152 AB EX   WPT
  1238 1773         504 S6=     0              ASSUME RAM
  1239 1774         314 ?S10=1                 ROM?
  1240 1775        1640 RTN NC                 RAM
  1241 1776         510 S6=     1              SAY ROM
  1242 1777        1740 RTN
*
  1244                        UNLIST
  1247                        END

  ERRORS :        0
```

SYMBOL TABLE

```
ALPHOP  1756   -
BNBCD3  1563   -   1572
CPABC    264   -    237
CPFKB   1266   -
CPYS6M  1770   -
GCHAR   1604   -
GCHR40  1614   -   1612
GCHR45  1623   -   1620
GCHR47  1633   -   1624
GCHR48  1636   -   1626
GCHR49  1640   -   1635
GCHR50  1641   -   1622
LBLCK   1717   -
LBLCK3  1754   -   1737 1727
LCDASC  1644   -   1632
LINELB  1573   -
LINELC  1575   -
LIST    1003   -
LISTN   1057   -   1011
LISTNB  1075   -
OUTPPS   257   -    225
OUTPPX   733   -    776
OUTPRP  1163   -   1105
PBINBC  1555   -
PBINBD  1556   -
PDER00   305   -    357
PDER10   322   -    312
PDER20   333   -    324
PDER50   336   -    310
PDER55   341   -    335   332   321   316
PDER90   360   -    355
PDEROW   300   -
PFK10   1313   -   1305 1301
PFK11   1324   -
PFK12   1331   -   1275
PFK17   1427   -   1330
PFK20   1432   -
PFK300  1331   -
PFK305  1344   -   1376
PFK310  1346   -   1335
PFK315  1372   -   1364
PFK320  1377   -   1351
PFK330  1407   -   1402
PFK334  1415   -   1426
PFK337  1422   -   1370
PFK34   1453   -   1503
PFK340  1423   -   1412
PFK35   1454   -   1452
PFK38   1460   -   1450
PFK45   1466   -   1371
PFK47   1500   -   1476
PFK50   1502   -   1465
PFK52   1504   -   1463 1422
PFK55   1514   -   1525
PFK57   1526   -   1516
PFK59   1541   -   1554
```

```
PFK70    1543    -    1540
PL        272    -     231
PLBL      400    -     466
PLBL0     377    -
PLBL3     405    -
PLCD10   1676    -    1714
PNUMBE   1560    -
PNUMBR   1561    -    1603
PPGMRS    521    -
PPGMST    522    -
PPGS05    523    -     520
PPGS10    525    -     531
PPGS20    544    -     542
PPGS25    552    -     545
PPGS30    556    -     553
PPGS32    557    -     550
PPGS33    563    -     551
PPGS34    566    -     555
PPGS35    567    -     536
PPGS37    577    -     575
PPGS60    613    -     611
PPGS65    615    -     600
PPGSNL    517    -
PPROM1    632    -
PPROM2    642    -
PPROMT    631    -
PPS120     40    -      31     24
PPS200    256    -
PPXROM    716    -
PR.END    475    -
PR0110    363    -     303
PR1010    117    -
PR1026    130    -     126
PR1314     64    -      16     15
PRMT20    645    -     702
PRMT30    657    -     654
PRMT40    665    -     660
PRMT45    666    -     664
PROW0      21    -       0
PROW09     52    -      11
PROW1      25    -       1
PROW10    104    -      12
PROW11     53    -      13
PROW12     56    -      14
PROW2      27    -       2
PROW3      36    -       3
PROW9     140    -      52
PRP      1015    -
PRPA15   1220    -    1257
PRPA20   1222    -    1213
PRPA40   1225    -    1221
PRPA45   1242    -    1263
PRPA48   1243    -    1233
PRPA49   1244    -    1236
PRPA50   1246    -    1217
PRPERR   1026    -    1031
PRPINT   1017    -
PRTLCD   1671    -
PRTP15   1037    -    1022
PRTP16   1044    -    1040
```

```
PRTP10   1047   -   1036 1034
PRTP20   1051   -   1043
PRTP30   1101   -   1162
PRTP40   1103   -   1100
PRTP50   1107   -
PRTP55   1115   -   1113
PRTP60   1135   -   1133
PRTP80   1153   -   1242 1146
PRTPAC   1174   -   1116
PRTPL    1147   -   1126
PRW010     23   -     55
PRW120    445   -
PRW122    467   -    455
PRW124    510   -    473  471
PRW4-8     32   -     10    7     6     5     4
PRW910    145   -    114   61
PRW911    146   -    144
PRW930    156   -
PRW933    161   -    137
PRW935    177   -    160  103
PRW936    201   -    176
PRW938    214   -    212
PRW940    226   -    205
PRW945    246   -    251
PRW950    252   -    271  266
PRW960    253   -    277  274  247
PSTR10    423   -    434
PSTR20    435   -    431
PSTRNG    415   -    410  376
PT        275   -    241
PTXROW    411   -
PXR10     717   -
PXR19     737   -    721
PXR20     740   -    723
PXR30     761   -    752
PXROM     704   -
REGASC   1652   -
SMAEC     267   -    233
SPCASC   1655   -   1645
```

```
EXTERNAL REFERENCES

ASRCH    1023
ASRCH    1024
BINBCD    217
BINBCD    220
BINBD0    602   1573
BINBD0    603   1574
BINBDC   1556
BINBDC   1557
BPROM      44     71    132    151    370    462   1417
BPROM      45     72    133    152    371    463   1420
CKANGR    674   1701
CKANGR    675   1702
CKANGL    423   1517
CKANGL    424   1520
CPBYTE    254    342    425    613    677    767   1521   1570
CPBYTE    255    343    426    614    700    770   1522   1571
CPCMHD   1051
CPCMHD   1052
CPYS6M    372    413    464
CPYS6M    373    414    465
DATAPR   1761
DATAPR   1762
ENCP00    361   1715
ENCP00    362   1716
ENLCD    1674
ENLCD    1675
EOLCR     563
EOLCR     564
EOLL      560   1075   1120   1151   1166   1246
EOLL      561   1076   1121   1152   1167   1247
EOLR     1144
EOLR     1145
ERRNE    1026
ERRNE    1027
FLINKP   1044
FLINKP   1045
FNSTS     537   1107   1163
FNSTS     540   1110   1164
GCHAR    1676
GCHAR    1677
GETPC     523   1041   1153
GETPC     524   1042   1154
GLINE#   1073
GLINE#   1074
GTBYT    1747
GTBYT    1750
GTRMAD    717
GTRMAD    720
INADXP    401    405
INADXP    402    406
INCAD      75    447   1743   1745
INCAD      76    450   1744   1746
IPRT     1067
IPRT     1070
LBLCK     533
LBLCK     534
```

```
LCDASC    670
LCDASC    671
LINELC    604
LINELC    605
NBYTAB    115     154     344     707
NBYTAB    116     155     345     710
NXBTXP    374     403     432
NXBTXP    375     404     433
NXTBYT     77     451     525
NXTBYT    100     452     526
OUTPPS     34     515     733    1344
OUTPPS     35     516     734    1345
PAD1+A   1142
PAD1+A   1143
PBINB0    763     772
PBINB0    764     773
PBLANK    325    1441    1543    1601
PBLANK    326    1442    1544    1602
PBYTDU   1704
PBYTDU   1705
PBYTEC   1711
PBYTEC   1712
PDEROW     25
PDEROW     26
PFK20     735
PFK20     736
PLBL      757
PLBL      760
PNUMBB    221
PNUMBB    222
PPGMRS   1127    1147
PPGMRS   1130    1150
PPGSNL   1201    1227
PPGSNL   1202    1230
PPROM1     42      67     147     366     460     513    1415
PPROM1     43      70     150     367     461     514    1416
PPROM2    730    1430
PPROM2    731    1431
PPROMT     32     130
PPROMT     33     131
PPS200   1437
PPS200   1440
PR1010   1405
PR1010   1406
PR3RT    1766
PR3RT    1767
PRKC     1552
PRKC     1553
PRQUOT    435    1504    1526
PRQUOT    436    1505    1527
PRTMSG    165     475     742    1237    1336    1352    1372
PRTMSG    166     476     743    1240    1337    1353    1373
PRTP50   1264
PRTP50   1265
PRW120     62
PRW120     63
PRW930     50    1456
PRW930     51    1457
PRW938   1500
PRW938   1501
```

```
PRX10    1172
PRX10    1173
PTXROW     17
PTXROW     20
PUTPCF   1053
PUTPCF   1054
PUTPCL   1157
PUTPCL   1160
PWAIT    1122   1260
PWAIT    1123   1261
PXR10    1311
PXR10    1312
PXROM     107
PXROM     110
SKPLIN   1155
SKPLIN   1156

End of VASM assembly
*************************************************************************
VASM ROM ASSEMBLY           REV.  6/81A

OPTIONS: L C S

    2                     FILE   SCPR4B
**************************************************************************
*
* FILLIN - FILL LINE WITH BLANKS AND PRINT
*
* USES: A.X, C.X, N, S9, AND TWO ADDITIONAL SUBROUTINE LEVELS
* IN: C=# OF LAST CHARACTER POSITION FILLED SO FAR
*      PT=0
* OUT: NOTHING
* ASSUMES: HEXMODE, S9=PRINTER INTERFACE ERROR FLAG
    12
*
* FILLNP - SETS THE POINTER TO 0 AND FALLS INTO FILLIN
*
    16                    ENTRY  FILLIN
    17                    ENTRY  FILLNP
    18     0 FILLNP 1634 PT=     0
    19     1 FILLIN  460 LDI
    20     2          30 CON     24
    21     3         406 A=C     X
    22     4         230 C=G
    23     5         706 A=A-C   X
    24                    LEGAL
    25     6           1 GOSUB   PAD
    25     7           0
    26    10           1 GOLONG EOLR
    26    11           2
*
*
**************************************************************************
*
* INADXP - INCREMENT ADDRESS, USING S6 TO DECIDE ROM/RAM
* USES: A3:0
* IN: A3:0=ADDRESS
*      S6=1 FOR ROM, S6=0 FOR RAM
*      PT=3
* OUT: A3:0 INCREMENTED TO NEXT BYTE ADDRESS
* ASSUMES: HEXMODE
*
                          ENTRY   INADXP
      12 INADXP   514 ?S6=1                       ROM?
      13            1 GOLNC   INCADA              NO
      14            2
      15          556 A=A+1                       YES
      16         1740 RTN

*
****************************************************************************
*********************  PRT1 --- PRINT X IN TRACE  ***************************
****************************************************************************
                          ENTRY   PXTR
6C0C  17 PXTR       1 GOSUB   CKTRCE              SEE IF PTR IN TRACE MODE
      20            0
      21         1740 RTN                         NO
      22            1 GOSUB   FNDPTR              LOOK FOR THE PRINTER
      23            0
      24         1740 RTN                         PRINTER NOT FOUND
      25          314 ?S10=1                      ROMFLAG?
      26           37 GOC     PXTR2  (   31) YES
      27         1514 ?S12=1                      PRIVACY?
      30           73 GONC    PXTR4  (   37) NO
      31 PXTR2   1314 ?S13=1                      RUNNING?
      32          107 GOC     PXTREX (   42) YES
      33            1 GOSUB   LDSST0              NO, PUT UP STATUS SET 0
      34            0
      35          114 ?S4=1                       SINGLE-STEPPING?
      36           47 GOC     PXTREX (   42) YES
      37 PXTR4      1 GOSUB   FNSTS
      40            0
      41          114 ?S4=1                       "ALL" MODE?
      42 PXTREX     1 GOLNC   UNL                 NOPE
      43            2
      44          240 SEL P
      45          214 ?S5=1                       SUPER TRACE ?
      46            1 GOLC    PRSTKX              YES, PRINT STACK
      47            3
      50            1 GOSUB   INITC
      51            0
* PXTR DROPS INTO PRXSUB HERE
*
* PRXSUB (PRINT X SUBROUTINE) - PRINT X WITH THREE STARS AND EOLR
*
* USES - THREE ADDITIONAL SUBROUTINE LEVELS!!!
*        A, B, C, P, Q, G, S0-S9
*
* INPUTS - S9 IS PRINTER INTERFACE ERROR FLAG
*          VALUE OF X IS IN R3
* OUTPUTS - ONE LINE TO PRINTER BUFFER, S9 ERROR FLAG
* ASSUMES - CHIP 0 ENABLED, HEX MODE
*
                          ENTRY   PRXSUB
      52 PRXSUB   370 C=REGN 3                    GET X REG
      53            1 GOSUB   ACXSUB
      54            0
      55            1 GOSUB   PRTMSG
      56            0
      57          244 CON     @244                4 BLANKS
```

```
   60              52 CON    052          *
   61              52 CON    052          *
   62             452 CON    0452         *
   63             433 GOTO   EOLREX ( 126) EOLR
*
**************************************************************************
********** PRT15 - SST/BST ***********************************************
**************************************************************************
                      ENTRY  XPRT15
   64 XPRT15      660 C=STK
   65             660 C=STK
   66             530 M=C                 SAVE SSTBST RTN IN M
   67               1 GOSUB  DATAPR
   70               0
   71             630 C=M
   72             560 STK=C
   73               1 GOLONG PR15RT
   74               2
   75             206 CON    0206         F
   76              25 CON    21           U
   77               2 CON    2            B
  100              22 CON    022          R
  101              20 CON    020          P
                      ENTRY  PRBUF
  102 PRBUF         1 GOSUB  CKEN
  103               0
  104            1740 RTN
  105               1 GOSUB  FNDPTR
  106               0
  107             633 GOTO   PECHKJ ( 172)
  110             404 S8=    0
  111               1 GOSUB  INADV
  112               0
  113               1 GOLONG LPECHK
  114               2
*
***************************************************************************
* THIS ENTRY IS USED BY TIMER TOO. IT REQUIRED USED ONLY A,C,N,PT
* S0-S7,S9 AND +2 SUB LEVEL
*+
                      ENTRY  PADV
  115 PADV          1 GOSUB  CKEN         SEE IF OK TO PRINT
  116               0
  117            1740 RTN                 NO
  120               1 GOSUB  FNDPTR       SEE IF PRINTER PRESENT
  121               0
  122             503 GOTO   PECHKJ ( 172) NO, GOTO DISPLAY ERROR MESSAGE
  123             404 S8=    0
  124               1 GOSUB  INADV        GET OUT OF COLUMN MODE IF IN
  125               0
  126 EOLREX        1 GOLONG RPECHK       NO, EOLR, CHECK PRINTER ERR
  127               2
****************************************************************************
  130             222 CON    0222         R
  131              10 CON    010          H
  132               3 CON    03           C
  133               3 CON    03           C
  134               1 CON    01           A
                      ENTRY  ACCHR
                      ENTRY  ACCHRX
```

```
 137  135 ACCHR      1 GOSUB  CX<128          X TO BINARY, RTN IF X<128
 137  136            0
 138  137 ACCHRX   206 B=A    X               SAVE A.X IN B.X
 139  140            1 GOSUB  IACHR
 139  141            0
 140  142          306 C=B    X               PUT THE CHAR INTO C.X
 141  143            1 GOSUB  CKANGB 7F52     CHECK IF THE ANGEL SIGN
 141  144            0
 142  145          406 A=C    X
 143  146          460 LDI
 144  147           12 CON    10
 145  150         1546 ? A#C  X               IS IT THE DIAMON ?
 146  151          167 GOC    PPECHK ( 167) NO
 147  152            6 A=0    X               DIAMON IS 0
 148  153          143 GOTO   PPECHK ( 167) A(X) TO PRINTER, CHECK ERRORS
*********************************************************************
*
* ACCOL - ACCUMULATE COLUMN IN PRINTER BUFFER
*
 153  154          214 CON    0214            L
 154  155           17 CON    15              O
 155  156            3 CON    3               C
 156  157            3 CON    3               C
 157  160            1 CON    1               A
 158                   ENTRY  ACCOL
 159  161 ACCOL      1 GOSUB  CX<128          "X" TO BINARY, CHECK < 128
 159  162            0
 160  163          206 B=A    X               SAVE A.X IN B.X
 161  164            1 GOSUB  IACOL           INITIALIZE COL OUT PRINT
 161  165            0
 162  166          146 AB EX  X               RESTORE A.X
 163
 164  167 PPECHK   246 AC EX  X
 165  170            1 GOSUB  PBYTEC          C(X) BIT PATTERN TO PRINTER
 165  171            0
 166  172 PECHKJ     1 GOLONG PECHK           ERROR CHK AND EXIT
 166  173            2
*
* CKTRCE - CHECK IF PRINTER IN TRACE MODE
* ASSUME: CHIP 0 ENABLE
* OUTPUT : CHIP 0 ENABLE
* RETURN TO P+1 IF :
*     1. PRINTER NOT EXIST
*     2. PRINTER NOT IN TRACE MODE AND RUNNING
* RETURN TO P+2 IF :
*     1. PRINTER PRESENT AND NOT RUNNING
*     2. PRINTER PRESENT AND RUNNIING AND PRINTER IN TRACE MODE
*
 178                   ENTRY  CKTRCE
*
 180  174 CKTRCE  1140 SETHEX
 181  175            1 GOSUB  LDSST0
 181  176            0
 182  177         1614 ?S0=1                  PRINTER PRESENT ?
 183  200         1640 RTN NC                 NO
 184  201         1314 ?S13=1                 RUNNING ?
 185  202           73 GONC   CKTRC1 ( 211) NO, RETURN TO P+2
 186  203          744 C=HPIL 7
 186  204          772
 186  205          703
```

```
206          1530 ST=C
207           114 ?S4=1                         PTR IN TRACE MODE ?
210          1640 RTN NC                        NO
211 CKTRC1      1 GOLONG RTNP+2
212             2
*
* PRSVC (PRINTER SERVICE) - I/O SERVICE ENTRY POINT LOGIC.
*
* FOR FLOWCHART SEE  BW'S LAB BOOK #8377 P.15
*
* ENTERS WITH SS0 UP.
* IF NORMAL RETURN TO RMCK10 IS MADE, C MUST BE PRESERVED AND
* SS 0 MUST BE UP.
*
                    ENTRY  PRSVC
* WHEN WE ARRIVE AT PRSVC, WE HAVE ALREADY CHECKED THAT THE PRINTER
* IS TURNED ON.
213 PRSVC     246 AC EX  X                      COPY PAUSETIMER TO C.X
214           530 M=C                           SAVE C IN M
215          1670 C=REGN 14
216          1074 RCR    2
217          1730 CST EX
220          1414 ?S1=1                         PKSEQ ?
221           407 GOC    PSVC90 ( 261) YES, IGNORE SERVICE REQUEST
222          1730 CST EX                        RESTORE SST 0
223             1 GOSUB  FNDPTR                 LOOK FOR THE PRINTER
224             0
225           253 GOTO   PSVC80 ( 252) PRINTER NOT FOUND
226          1114 ?S9=1                         INTERFACE ERROR?
227           327 GOC    PSVC90 ( 261) YES
230            14 ?S3=1                         OUT OF PAPER?
231            57 GOC    PSVC10 ( 236) YES
232          1614 ?S0=1                         OUT OF PAPER HOLD?
233           123 GONC   PSVC30 ( 245) NO
234          1414 ?S1=1                         PRINT KEY DOWN ?
235            47 GOC    PSVC20 ( 241) YES, SEND EOL, DO SHORT ADV
236 PSVC10      1 GOSUB  OOPMSG                 DISPLAY "OUT OF PAPER"
237             0
240           213 GOTO   PSVC90 ( 261)
241 PSVC20      1 GOSUB  PRBUF
242             0
243             1 GOLONG ADV50
244             2
245 PSVC30   1414 ?S1=1                         PRINT KEY DOWN?
246           277 GOC    PKEY   ( 275) YES
247          1014 ?S2=1                         NO, ADV KEY DOWN?
250             1 GOLC   ADVKEY                 YES
251             3

252 PSVC80   1670 C=REGN 14
253           274 RCR    5
254          1530 ST=C
255            14 ?S3=1                         IN MANUAL MODE ?
256            33 GONC   PSVC90 ( 261) NO
257           344 HPL=CH 3
260             1 CH=    @000                   SHUT OFF AUTO IDY
261 PSVC90   1670 C=REGN 14                     RESTORE SS0 TO ST
262          1530 ST=C
263 PSVC95      1 GOSUB  UNL
264             0
```

```
 265            630 C=M                      RESTORE C
 266            406 A=C      X               RESTORE PAUSETIMER
 267            646 A=A-1    X               ADJUST PAUSETIMER
 270             23 GONC     PSVC99 ( 272)
 271              6 A=0      X               DON'T ALLOW PSETMR TO ROLL OVER
 272 PSVC99    1104 S9=      0
 273              1 GOLONG RMCK10
 274              2
*
* PKEY - SERVICE PRINT KEY
*
 275 PKEY      1530 ST=C                     RESTORE STATUS SET 0
 276             14 ?S3=1                    PROGRAM MODE?
 277             23 GONC     PKEY15 ( 301) NO
                                             PROGRAM MODE
 300            110 S4=      1               SET INSERT BIT FOR
* DSPLN+ AND NLT040. OVERLAYS SSTFLAG IN SS 0
 301 PKEY15       1 GOSUB    DSPLN+
 302              0
 303              1 GOSUB    MESSL
 304              0
 305             20 CON      16              P
 306             22 CON      18              R
 307           1030 CON      @1030           X
 310           1214 ?S7=1                    ALPHAMODE?
 311             53 GONC     PRT30  ( 316) NO

 312           1670 RABCR                    SCRAP THE X
 313              1 GOSUB    MESSL           ADD "A" TO GET "PRA"
 314              0
 315           1001 CON      @1001           A
 316 PRT30        1 GOSUB    LEFTJ
 317              0
 320              1 GOSUB    ENCP00
 321              0
 322            134 PT=      4               SET UP FC FOR PRA OR PRX
 323           1220 LC       10              FC FOR PRX=A754
 324            720 LC       7
 325            520 LC       5
 326            420 LC       4               ASSUME PRX
 327           1214 ?S7=1                    ALPHAMODE?
 330             43 GONC     PKEY35 ( 334) NO
 331           1034 PT=      2               YES, FC FOR PRA=A748
 332            420 LC       4
 333           1020 LC       8
 334 PKEY35     530 M=C                      FC TO M[4:1]
 335           1630 C=ST                     COPY ST TO G FOR NLT040
 336           1634 PT=      0
 337            130 G=C

 340            460 LDI
 341             70 CON      @70             INITIALIZE TIMER
 342 PRT40     1146 C=C-1    X
 343            177 GOC      PRT60  ( 362) TIMEOUT
 344            346 BC EX    X               SAVE TIMER IN B.X
 345              1 GOSUB    FNSTS           GET PRINTER STATUS
 346              0
 347            306 C=B      X               TIMER BACK TO C.X
 350           1114 ?S9=1                    PRINTER ERROR?
 351             37 GOC      PRT50  ( 354) YES, ASSUME KEY IS UP.
```

```
352        1414 ?S1=1                PRINT KEY STILL DOWN?
353        1677 GOC    PRT40  ( 342) YES
*SINCE THE PRINT KEY WON'T BE RECOGNIZED UNTIL THE PRINTER IS IDLE
*AGAIN, AND SINCE THE PRINTER KEYBOARD DOESN'T LATCH KEYS, THE TIME
*TAKEN TO PRINT IS USED TO DEBOUNCE THE KEY.
                ENTRY  PRT50

354 PRT50     1 GOSUB  UNLRSF
355           0
356         630 C=M
357         416 A=C                  FC BACK TO A[4:1]
360           1 GOLONG NLT040
361           2

    PRT60                            NULL OUT THE PRINT KEY
362         404 S8=    0
363           1 GOSUB  MSGA          "NULL" MESSAGE TO DISPLAY
364           0
365           0 XDEF   MSGNL
366         253 GOTO   ADV02  ( 413)
*
* ADVKEY - SERVICE PAPER ADVANCE KEY
*
                ENTRY  ADVKEY
367 ADVKEY  404 S8=    0             PREPARE TO GET OUT OF COL MODE
370        1574 RCR    12
371        1730 CST EX               GET BACK 2ND STS BYTE
372         114 ?S4=1                ALREADY IN SPEC-K MODE ?
373          47 GOC    ADVCKC ( 377) YES, SEE IF COL MODE
374           1 GOSUB  SPEC-K        SELECT SPEC-K MODE
375           0
376          43 GOTO   ADV01  ( 402)
377 ADVCKC 1414 ?S1=1                IN COL. OUT MODE ?
400           1 GSUBC  INITSM        YES, IF S1=1
401           1
402 ADV01  1670 C=REGN 14
403        1530 ST=C
404          14 ?S3=1                IN PROG MODE ?
405         177 GOC    ADV04  ( 424) YES
406           1 GOSUB  DATAPR        PRINT DATA ENTRY STRING
407           0
410           1 GOSUB  EOLR          EOLR= GET OUT ANY PARTIAL LINE
411           0
412        1104 S9=    0             IGNORE ANY ERROR SO FAR
413 ADV02     1 GOSUB  FNSTS
414           0
415        1114 ?S9=1                PRINTER ERROR?
416         557 GOC    ADV50  ( 473) YES, ASSUME KEY IS UP
417        1014 ?S2=1                ADV KEY STILL DOWN?
420        1737 GOC    ADV02  ( 413) YES
421        1414 ?S1=1                PRINT KEY STILL DOWN ?
422        1717 GOC    ADV02  ( 413) YES
423 ADV03   503 GOTO   ADV50  ( 473) NO. ALL DONE.

* SEND COMMAND TO HELIOS TO IGNORE LOCAL PAPER ADVANCE
424 ADV04     1 GOSUB  PRTMSG        PROGRAM MODE, IGNORE PAPER ADVANC
425           0
426         777 CON    @777          IGNORE PAPER ADVANCE COMMAND
427         110 S4=    1             SET S4 FOR DSPLN+,ETC.
430           1 GOSUB  DSPLN+
431               0
432               1 GOSUB  MESSL
433               0
434               1 CON    1                A
435               4 CON    4                D
436            1026 CON    @1026            V
437               1 GOSUB  LEFTJ
440               0
441               1 GOSUB  ENCP00
442               0
443             134 PT=    4
444            1020 LC     8                FC FOR
445            1720 LC     15                ADV
446             530 M=C
447            1630 C=ST
450            1634 PT=    0
451             130 G=C
452             460 LDI
453              70 CON    @70
454 ADV10      1146 C=C-1  X                TIMEOUT?
455             137 GOC    ADV30    ( 470)  YES
456             346 BC EX  X
457               1 GOSUB  FNSTS
460               0
461            1114 ?S9=1                   PRINTER ERROR?
462              47 GOC    ADV20    ( 466)  YES, ASSUME KEY IS UP
463             306 C=B    X
464            1014 ?S2=1                   ADV KEY STILL DOWN?
465            1677 GOC    ADV10    ( 454)  YES
466 ADV20         1 GOLONG PRT50
467               2

    ADV30                                   DO LOCAL PAPER ADVANCE
470            1146 C=C-1  X                C(X): FF TO FE
                    LEGAL
* SEND HELIOS A COMMAND TO RE-ENABLE LOCAL PAPER ADVANCE
471               1 GOSUB  PBYTEC
472               0
                    ENTRY  ADV50
473 ADV50         1 GOSUB  UNLRSF
474               0
475               1 GOLONG ABTS10
476               2
*
*
*****************************************************************************
*
* BLDSPEC - BUILD SPECIAL CHARACTER
*
477             203 CON    @203             C
500               5 CON    5                E
501              20 CON    16               P
502              23 CON    19               S
503               4 CON    4                D
504              14 CON    12               L
505               2 CON    2                B
                    ENTRY  BLDSPC
506 BLDSPC        1 GOSUB  CX<128           "X" TO BINARY, CHECK <128
507               0
510             270 C=REGN 2                GET Y
```

```
511        1176 C=C-1   S
512        1376 ? C#0   S                  IS Y A NUMBER?
513          23 GONC    BLD10   ( 515) NO
514         116 C=0                        YES. INITIALIZE TO NULL STRING
515 BLD10   756 C=C+C
516         756 C=C+C
517        1574 RCR     12
520        1334 PT=     13
521         120 LC      1                  LEAVE SIGN DIGIT= 1= ALPHA DATA
522        1712 C SR    WPT                LEAVE 3 MSB OF DIGIT 12= 000
523         752 C=C+C   WPT
524        1006 C=A+C   X
525         356 BC EX
526           1 GOLONG  DROPST
527           2
*
* ACSPEC - ACCUMULATE SPECIAL CHARACTER
*
* USES A,C,M,N,PT,S9:0, & 2 ADDITIONAL SUBROUTINE LEVELS
*
*-ACSPCC= ACCUMULATE SPECIAL CHARACTER IN C REGISTER
*-USES:    A,C,M,N,      PT,  S0-S9,  2 ADDITIONAL SUB LEVELS
*-INPUTS:  C= SPECIAL CHARACTER,  CHIP 0 ENABLED
*      RTNS WITH CHIP 0 ENABLED
*
530          203 CON    @203           C
531            5 CON    5              E
532           20 CON    16             P
533           23 CON    19             S
534            3 CON    3              C
535            1 CON    1              A
                 ENTRY  ACSPEC
536 ACSPEC   370 C=REGN 3
537         1176 C=C-1  S
540         1176 C=C-1  S
541 AERRDE     1 GOLNC  ERRDE
542            2
543            1 GOSUB  IACOL          INITIALIZE COL OUT PRINT
544            0
545         1334 PT=    13
546          620 LC     6
547          436 A=C    S
550          370 C=REGN 3
551          210 S5=    1              EXIT TO PECHK
552           33 GOTO   ACSPCC ( 555)

553 SPEC10   630 C=M
554          756 C=C+C
                 ENTRY  ACSPCC
555 ACSPCC  1374 RCR    13
556          756 C=C+C
557          756 C=C+C
560          530 M=C
561         1574 RCR    12
562            1 GOSUB  PBYTDU
563            0
564          676 A=A-1  S              DONE WITH REGISTER YET?
565         1663 GONC   SPEC10 ( 553) NO
566          214 ?S5=1                 EXIT TO PECHK ?
567         1640 RTN NC                NO JUST RETURN
```

```
*
*ACSPEC FALLS INTO PECHK HERE!!!!!!!!!!!!!!
*
* PECHK (PRINTER ERROR CHECK) - IF S9=0 THEN DOES AN IMMEDIATE RETURN
* ELSE FALLS INTO PEDIAG
*
* PEDIAG (PRINTER ERROR DIAGNOSTIC) - PRODUCES MOST APPROPRIATE ONE
* OF THE POSSIBLE PRINTER ERRORS.  EXITS TO MAINFRAME ERROR ROUTINE.
*
6D27                      ENTRY   PECHK
                          ENTRY   PEDIAG
 570 PECHK  1114 ?S9=1                      ANY PRINTER ERROR?
 571           1 GOLNC    UNLRSF            NOPE
 572           2

 573 PEDIAG    1 GOSUB    FNDPTR            SEE IF PRINTER IS THERE
 574           0
 575         243 GOTO     PE10    ( 621)    PRINTER NOT FOUND
 576          14 ?S3=1                      OUT-OF-PAPER?
 577         243 GONC     PE30    ( 623)    NO, SOME OTHER ERROR
 600           1 GOSUB    OOPMSG            YES
 601           0
 602         153 GOTO     PE05    ( 617)
 603 NOPTR     1 GOSUB    PLEREX            NO
 604           0
 605          16 CON      @16               N
 606          17 CON      @17               O
 607          40 CON      @40
 610          20 CON      @20               P
 611          22 CON      @22               R
 612          11 CON      @11               I
 613          16 CON      @16               N
 614          24 CON      @24               T
 615           5 CON      @05               E
 616        1022 CON      @1022             R
 617 PE05      1 GOLONG ERRRTN
 620           2

6C0
 621 PE10   1114 ?S9=1
 622        1613 GONC     NOPTR   ( 603)
 623 PE30      1 GOLONG PILERR
 624           2
*
 625 UNLEX     1 GOLONG UNL
 626           2
*
                          FILLTO @627
 627        0000 NOP
*
*-INITSC= MODE TO PRINTER (SPECIAL CHARACTER)
*-INITSM= INITIALIZE - SEND MODE TO PRINTER
*
*-USES:   C,N,  S8, S9 FOR ERRORS,  PT,  NO ADDITIONAL SUB LEVELS
*-INPUTS: S8=1 FOR COLUMN OUT MODE,  ELSE S8=0,    HEXMODE
*-OUTPUTS: CHIP 0 ENABLED, HEXMODE
*
                          ENTRY   INITSC
                          ENTRY   INITSM
 630 INITSC  410 S8=      1                 COLUMN OUT MODE
 631 INITSM  106 C=0      X                 ENABLE CHIP 0
```

```
   502  632         1160 DADD=C
   503  633          334 PT=     10
   504  634          753 GOTO    INIT12 ( 731) SEND MODE TO PRINTER
*
*
   507
*
* IPRT - INITIALIZE ORDINARY PRINTING FCNS (PRTX, ETC)
* 1. CALL CKEN.  IF RETURN IS TO P+1 THEN POP THE SUBROUTINE STACK
*    AND RTN.
* 2. CALL FNSTS
* 3. CALL OOPCHK
* 4. FORCE OUT ANY PARTIAL LINE.
* 5. SEND MODE IF NECESSARY
*
* SOMETIMES DOES A 2 LEVEL RETURN!
* USES: C, N, S0-S9, PT, AND 1 ADDITIONAL SUBROUTINE LEVEL
* INPUT: NONE
* OUTPUT: S9 IS THE PRINTER INTERFACE ERROR FLAG
* ASSUMES: HEXMODE, CHIP 0 ENABLED
*
*
* IPRTM - INITIALIZE PRINT FOR MAINFRAME PRINTING FCNS VIEW AND AVIEW
* SAME AS IPRT EXCEPT CALLS CKOEN INSTEAD OF CKEN.
*
* IACHR - INITIALIZE ACCUMULATE CHARACTER FCNS.  SAME AS IPRT EXCEPT
* DOESN'T FORCE OUT PARTIAL LINES AND USES 2 ADDITIONAL SUBROUTINE
* LEVELS.
*
* IACOL - INITIALIZE ACCUMULATE COLUMN FCNS.  SAME AS IACHR EXCEPT
* SETS UP COL OUT MODE INSTEAD OF CHARACTER OUT MODE.  NOTE IACHR'S
* USE OF SUBROUTINE LEVELS.
*
* IAUNA
* [redacted] - INITIALIZE AUTOMATIC PRINT FCNS WHICH PRINT IN BOTH "NORM"
* AND "ALL" PRINTER MODES.  SIMILAR TO IPRT EXCEPT HAS DIFFERENT RETURNS
* AND LOOKS AT PRINTER MODES INSTEAD OF CALLING CKEN.
*
* RETURNS TO P+1 IF NO PRINTING
* RETURNS TO P+2 IF PRINTING IS OK
* USES: C, N, S0-S9, PT, AND 1 ADDITIONAL SUBROUTINE LEVEL
* INPUT:NONE
* OUTPUT: S9 IS THE PRINTER INTERFACE ERROR FLAG
* ASSUMES: HEXMODE, CHIP 0 ENABLED
*
* IAUALL
* [redacted] - INITIALIZE AUTOMATIC PRINT FCNS WHICH PRINT IN "ALL" MODE ONLY.
* SAME AS IAUNA EXCEPT RETURNS TO P+1 WHEN PRINTER IS IN NORMAL MODE, AND
* INPUT REQUIRES S8=0.
*
* FLOWCHARTS FOR PRECEDING INITIALIZE ROUTINES ARE IN DRC'S LAB
* BOOK #8364 P.46
*
*
* INITC (INITIALIZE COMMON PATH) - SPECIAL ENTRY POINT FOR PRT1 AND PRT2
* LOGIC WHICH OPTIMIZES SPEED WHEN NO PRINTING IS DESIRED.
*
* USES: C, N, S0:9, PT, & 1 ADDITIONAL SUBROUTINE LEVEL
* IN: S9=PRINTER INTERFACE ERROR FLAG
*     C13:12=2ND BYTE OF PRINTER STATUS
*     S7:0=1ST BYTE OF PRINTER STATUS
* OUT: S9=PRINTER INTERFACE ERROR FLAG
* ASSUMES: HEXMODE, CHIP 0 ENABLED
*
*
* INIT5 - SPECIAL ENTRY POINT FOR PRT5
* SAME AS INITC EXCEPT FOR INPUT.
* IN: S9=PRINTER INTERFACE ERROR FLAG
*     B[13:12] = 2ND BYTE OF PRINTER STATUS
*     B[1:0] = 1ST BYTE OF PRINTER STATUS
*
                         ENTRY  IPRT
                         ENTRY  IACHR
                         ENTRY  IACOL
                         ENTRY  IAUNA
                         ENTRY  IAUNB
                         ENTRY  IAUALL
                         ENTRY  INITC
                         ENTRY  INIT5
6D9D  635 IPRT       1 GOSUB  CKEN 6FB9      OK TO PRINT?  OLD PRINTER : RTN !!
      636            0
      637           53 GOTO   IN999  ( 644) P+1 - NO  No [illegible] enabled Flag [illegible] Clear
                                            P+2 - YES
      640            1 GOSUB  FNDPTR 747D
      641            0
      642         1573 GOTO   PE10   ( 621) DISPLAY ERROR MESSAGE  No PTR found
      643          373 GOTO   INITC  ( 702) 6DC2

6DA4  644 IN999     40 SPOPND XQ→GO
      645         1740 RTN

                         ENTRY  INADV
          IACHR
6DA6  646 [redacted]     404 S8=    0          SET UP FOR CHAR OUTPUT
      647            1 GOSUB  CKEN           OK TO PRINT?
      650 IN20       0
      651         1733 GOTO   IN999  ( 644) P+1 - NO
      652            1 GOSUB  FNDPTR
      653            0
      654         1453 GOTO   PE10   ( 621) NOT FOUND, DISPLAY ERROR MESSAGE
      655 INADV      1 GOSUB  OOPCHK         P+2 - YES
      656            0
      657          363 GOTO   INIT10 ( 715)
          IACOL
      660 [redacted]     410 S8=    1          SET UP FOR COL OUTPUT
      661         1663 GOTO   IN20   ( 647)
*         IAUNA
      662 [redacted]     410 S8=    1          NORM MODE IS OK
*
                         FILLTO 0662
*
* IAUALL CALLED BY TIMER ROM TOO. IT REQUIRED USE ONLY A,C,N
* S0-S7,S9,PT AND +2 SUB LEVEL     6DB3
*         IAUALL
      663 [redacted]       1 GOSUB  FNDPTR         LOOK FOR PRINTER
      664            0
      665         1740 RTN                   PRINTER NOT FOUND
      666 IAUNB    114 ?S4=1                 "ALL" MODE?
      667           57 GOC    IN40   ( 674) YES, SO PRINT
      670          414 ?S8=1                 PRINT IN NORM MODE?
```

```
  616  671         1343 GONC    UNLEX  ( 625) NO
  617  672          214 ?S5=1                 NORM MODE?
  618  673         1323 GONC    UNLEX  ( 625) NO, SO DON'T PRINT.
  619  674 IN40     660 C=STK                 INC RTN ADDR
  620  675         1072 C=C+1   M
  621  676          560 STK=C
  622  677           33 GOTO    INITC  ( 702)
  623
  624  700 INIT5    316 C=B                   RESTORE STATUS TO C
  625  701         1530 ST=C                   AND S7:0
  626  702 INITC      1 GOSUB   OOPCHK
  626  703            0
  627  704          404 S8=     0             COL OUT NOT DESIRED
  628  705         1214 ?S7=1                 EOLL?
  629  706           77 GOC     INIT10 ( 715) YES
  630  707         1414 ?S1=1                 IN COL OUT MODE ?
  631  710            1 GSUBC   INIT60        YES, GET OUT OF COL OUT MODE
  631  711            1
  632  712          214 ?S5=1                 BUFFER EMPTY ?
  633  713            1 GSUBNC EOLCR          NO,FORCE OUT PARTIAL LINE
  633  714            0
  634  715 INIT10 1670 C=REGN 14
* FLAG 12 (DIGIT 10 BIT 3) FOR DOUBLE WIDE
* FLAG 13 (DIGIT 10 BIT 2) FOR LOWER CASE
  637  716          334 PT=     10
  638  717          114 ?S4=1                 HELIO CHAR SET ?
  639  720          127 GOC     INIT15 ( 732) YES
*
  641                   ENTRY   SPEC-K
*
  643  721 SPEC-K   460 LDI
  644  722           33 CON     27            SEND "ESC <" TO GO INTO HELIO MOD
  645  723            1 GOSUB   PBYTEC
  645  724            0
  646  725          460 LDI
  647  726          174 CON     124
  648  727            1 GOSUB   PBYTEC
  648  730            0
  649  731 INIT12   263 GOTO    INIT60 ( 757)
  650  732 INIT15   742 C=C+C   PT            NUT DOUBLE WIDE?
  651  733           43 GONC    INIT20 ( 737) NO
  652                                         NUT DOUBLE WIDE
  653  734         1014 ?S2=1                 HELIOS DWM?
  654  735          223 GONC    INIT60 ( 757) NO. GO SEND MODE
  655  736           33 GOTO    INIT30 ( 741)
  656                                         NUT NOT DOUBLE WIDE
  657  737 INIT20 1014 ?S2=1                  HELIOS DWM?
  658  740          177 GOC     INIT60 ( 757) YES. GO SEND MODE
  659
  660  741 INIT30   742 C=C+C   PT            NUT LOWER CASE?
  661  742           43 GONC    INIT35 ( 746)
  662                                         YES, NUT LOWER CASE
  663  743         1614 ?S0=1                 HELIOS LOWER CASE?
  664  744          133 GONC    INIT60 ( 757) NO. GO SEND MODE
  665  745           33 GOTO    INIT40 ( 750)
  666      INIT35                             NUT NOT LOWER CASE
  667  746         1614 ?S0=1                 HELIOS LOWER CASE?
  668  747          107 GOC     INIT60 ( 757) YES. GO SEND MODE
  669
  670  750 INIT40   414 ?S8=1                 NUT COLUMN OUT?
```

```
 751             43 GONC   INIT50 ( 755) NO
                                         YES, NUT COLUMN OUT
 752           1414 ?S1=1                HELIOS SCOM?
 753           1540 RTN C                YES, RETURN
 754             33 GOTO   INIT60 ( 757) NO. GO SEND MODE
      INIT50                             NOT NUT COLUMN OUT
 755           1414 ?S1=1                HELIOS SCOM?
 756           1640 RTN NC               NO, RETURN
                    ENTRY  INIT60

*-INIT60-   SEND MODE COMMAND
*
*-USES:     C,N,        PT,        S8-S9,      NO ADDITIONAL SUB LEVELS
*-INPUTS:   S8=1 FOR COLUMN OUT, ELSE S8=0
*           PT= 10,   CHIP 0 ENABLED,    HEX MODE
*-OUTPUTS:  CHIP 0 ENABLED,  HEX MODE
*
 757 INIT60     334 PT=    10            SEND MODE COMMAND
 760           1670 C=REGN 14
 761            460 LDI
 762            330 CON    @330
 763           1730 CST EX
 764            742 C=C+C  PT            DWM?
 765             23 GONC   INIT70 ( 767) NO
 766           1010 S2=    1             YES, SET DWM
 767 INIT70     742 C=C+C  PT            LOWER CASE
 770             23 GONC   INIT80 ( 772) NO
 771           1610 S0=    1             YES, SET LCA
 772 INIT80     414 ?S8=1                COLUMN OUT?
 773            543 GONC   PBYTCS (1047) NO
 774           1410 S1=    1             YES, SET SCOM
 775            523 GOTO   PBYTCS (1047)
*
*************************************************************************
*
* PRKC - PRINT KEYCODE
* USES: A.M, C, N, S3, PT, AND 1 ADDITIONAL SUBROUTINE LEVEL
* IN: S7:0=KEYCODE, A(M)= CHARACTER COUNTER
* OUT: "RC" OR "-RC" TO PRINTER (R=ROW#, C=COL#)
*      A.M=A.M+#OF CHARS SENT TO PRINTER
* ASSUMES: HEXMODE, CHIP 0 ENABLED, S9=PRINTER INTERFACE ERROR FLAG
*
                    ENTRY  PRKC20
                    ENTRY  PRKC
 776 PRKC        14 ?S3=1                SHIFTED?
 777             73 GONC   PRKC10 (1006) NO
1000              4 S3=    0             YES
1001            460 LDI
1002             55 CON    @55           "-"
1003              1 GOSUB  CPBYTE
1004              0
* CAN'T USE PRTMSG HERE BECAUSE NOT ENOUGH SUBROUTINE LEVELS
1005            572 A=A+1  M             COUNT THE CHAR
1006 PRKC10     572 A=A+1  M             COUNT 2 MORE CHARS
1007            572 A=A+1  M
1010           1630 C=ST
1011           1434 PT=    1
1012            320 LC     3
1013              1 GOSUB  PRKC20        INCREMENT & SEND ROW
1014              0
```

```
    1015          1374 RCR     13                 "3" TO C.XS
    1016          1630 C=ST
    1017          1474 RCR     1                  ROW TO C.S
    1020          1176 C=C-1   S
    1021          1176 C=C-1   S
    1022          1176 C=C-1   S
    1023          1176 C=C-1   S                  "ENTER" ROW?
    1024            33 GONC    PRKC20 (1027) NO
    1025          1342 ? C#0   PT                 KEY#"ENTER"?
    1026            27 GOC     CPBYTE (1030) NOT "ENTER"
    1027 PRKC20   1042 C=C+1   PT
*
*PRKC FALLS INTO CPBYTE HERE!!!!!
*
2* PBYTEC - SEND A CONTROL BYTE TO THE PRINTER
*
* ON ENTRY, C[1:0]=BYTE TO BE SENT TO THE PRINTER
*     AND S9=ERROR FLAG
* USES: N, NO PT, S9 FOR ERRORS, NO ADDITIONAL SUB LEVELS
* IF S9=1 THEN DOES AN IMMEDIATE RETURN
* WAITS UP TO 1 SECOND FOR THE PRINTER TO BE NOT BUSY.  ON A TIMEOUT,
* SETS S9 AND RETURNS.
*
* PBYTDU - PRINT A BYTE OF DATA UNCONDITIONALLY.  SAME AS PBYTEC
* EXCEPT CLEARS BIT 7 OF THE DATA FRAME BEFORE SENDING IT TO THE
* THE PRINTER.
*
* CPBYTE - CONDITIONALLY PRINT BYTE.  LOOKS AT FLAG 55 BEFORE DROPPING
* INTO PBYTEC.  IF FLAG 55 IS CLEAR, THEN DOES AN IMMEDIATE RETURN
* WITHOUT SENDING ANYTHING TO THE PRINTER.  USED FOR COUNTING
* CHARACTERS TO SEE WHETHER THEY WILL FIT ON A LINE.  FLAG 55 IS THE
* PRINTER EXISTENCE FLAG, WHICH IS NOMINALLY ON ALL THE TIME THE
* PRINTER IS PLUGGED IN.
*
*
                           ENTRY   PBYTEC
                           ENTRY   PBYTDU
                           ENTRY   CPBYTE
    1030 CPBYTE    160 N=C
    1031           106 C=0     X
    1032          1160 DADD=C
    1033          1670 C=REGN 14
    1034          1730 CST EX
    1035          1614 ?S0=1                      FLAG 55?
    1036            47 GOC     CPBYT1 (1042) YES, SEND BYTE TO PRINTER
    1037          1730 CST EX                     NO, DON'T PRINT
    1040 PBYT01    260 C=N                        RESTORE C REGISTER
    1041          1740 RTN
    1042 CPBYT1   1730 CST EX
    1043           260 C=N
    1044            43 GOTO    PBYTEC (1050)

    1045 PBYTDU   1730 CST EX
    1046          1204 S7=     0                  SUPPRESS 8TH BIT
    1047 PBYTCS   1730 CST EX
6E28 1050 PBYTEC  1114 ?S9=1                      ANY ERROR SO FAR
    1051          1540 RTN C                      YES, RETURN IMMEDIATELY
    1052           160 N=C                        SAVE C IN N
    1053          1374 RCR     13                 CHECK IF IT IS A CMD BYTE ?
    1054           766 C=C+C   XS                 MSB SET
```

```
1055            123 GONC   PBYT05 (1067) NO, JUST AN ASCII
1056            644 C=HPIL 6
1057            672
1060            603
1061           1166 C=C-1  XS
1062           1046 C=C+1  X              TALKING TO A T.V. ?
1063             43 GONC   PBYT05 (1067) NO
1064            460 LDI
1065             40 CON    @40            REPLACE THE CMD WITH A BLANK
1066             23 GOTO   PBYT06 (1070)
1067 PBYT05     260 C=N
1070 PBYT06     144 HPL=CH 1              WRITE DATA CONTROL BITS
1071              5 CH=    @001
1072           1200 HPIL=C 2              SEND THE BYTE OUT
1073            106 C=0    X
1074 PBYT10     354 ORAV?  X
1075             77 GOC    PBYT12 (1104)
1076              0 NOP
1077              0 NOP
1100           1046 C=C+1  X
1101           1733 GONC   PBYT10 (1074)
1102 PBYT11       1 GOLONG RDFMER
1103              2
1104 PBYT12    1154 FRNS?
1105           1333 GONC   PBYT01 (1040)  RESTORE C
1106           1743 GOTO   PBYT11 (1102)

**********************************************************************
*
* PAD - SEND PRINTER A COMMAND TO SKIP THE NUMBER OF CHARS IN A.X
*
* USES: C.X, N, S9
* IN: A.X = # OF PADS DESIRED (0-23)
* OUT: NOTHING
* ASSUMES: HEXMODE, S9=PRINTER INTERFACE ERROR FLAG

* PAD1+A - ADDS ONE TO A.X AND DROPS INTO PAD

                    ENTRY  PBYA+C
                    ENTRY  PAD
                    ENTRY  PAD1+A
1107 PAD1+A     546 A=A+1  X
1110 PAD        460 LDI
1111            240 CON    @240
1112 PBYA+C    1006 C=A+C  X
                    LEGAL
1113           1353 GOTO   PBYTEC (1050)

*
**********************************************************************
***************** PRT6 -- PRINT MESSAGE ******************************
**********************************************************************
*
* USES: A,C,G,N,S8, AND 1 ADDITIONAL SUBROUTINE LEVEL
*
* INPUT: CONTENTS OF LCD REGISTERS
* OUTPUT: ONE LINE TO PRINTER
* ASSUMES: S8=1 ON ENTRY     RETURNS S8=1 ON EXIT.
*     ASSUMES ADDRESS OF MSG110 IN MAINFRAME IS ON THE TOP OF THE
*     OF THE SUBROUTINE STACK ON ENTRY  RETURNS WITH A GOLONG TO
*     MSG110 ON EXIT.
*
                      ENTRY   PMESSG
     1114 PMESSG 1534 PT=     12             SAVE S9 IN A[12]
     1115           2 A=0     PT
     1116        1114 ?S9=1
     1117          23 GONC    PMSG10 (1121)
     1120         542 A=A+1   PT
     1121 PMSG10  202 B=A     PT
     1122        1634 PT=     0              SAVE S7-S0 IN G
     1123        1630 C=ST
     1124         130 G=C
     1125          40 SPOPND                 FREE UP A SUBROUTINE LEVEL
     1126           1 GOSUB   FNDPTR (70AF)  LOOK FOR THE PRINTER
     1127           0
     1130         123 GOTO    PMSG16 (1142) PRINTER NOT FOUND
     1131           1 GOSUB   IAUNB (6DB6)
     1132           0
     1133          53 GOTO    PMSG15 (1140) P+1 - DON'T PRINT
                      FILLTO @1133           P+2 - PRINT
* TIMER ROM JUMP INTO HERE TO ITS ALARMS
*
6E5C 1134 TMRMSG    1 GOSUB   PRTLCD (6BB9)
     1135           0
     1136           1 GOSUB   EOLL (67FF)    SEND EOLL
     1137           0
     1140 PMSG15    1 GOSUB   UNL (704F)
     1141           0
     1142 PMSG16 1104 S9=     0              RESTORE S9
     1143        1534 PT=     12
     1144        1302 ?B#0    PT
     1145          23 GONC    PMSG20 (1147)
     1146        1110 S9=     1
     1147 PMSG20 1634 PT=     0              RESTORE S0-S7
     1150         230 C=G
     1151        1530 ST=C
     1152         410 S8=     1              RETURN S8=1
     1153           1 GOLONG MSG110
     1154           2
                      EJECT
```

```
*
*************************************************************
*PRFLAG-PRINT FLAGS AND STATUS INCLUDING SIZE,SIGMA
* LOCATION, TRIG MODE AND DISPLAY SETTING.
*************************************************************
   885 1155              223 CON    @223
   886 1156                7 CON    @7
   887 1157                1 CON    @1
   888 1160               14 CON    @14
   889 1161                6 CON    @6
   890 1162               22 CON    @22
   891 1163               20 CON    @20
   892                       ENTRY  PRFLAG
   893 1164 PRFLAG         1 GOSUB  IPRT (6D90)    INITIALIZE PRINT
   893 1165                0
   894 1166                1 GOSUB  PRTMSL (640   PRINT:LF,STATUS:,LF,SIZE=
   894 1167                0
   895 1170             1015 CON    @1015               CR
   896 1171               12 CON    @12                 LF
   897 1172              123 CON    @123          S
   898 1173              124 CON    @124          T
   899 1174              101 CON    @101          A
   900 1175              124 CON    @124          T
   901 1176              125 CON    @125          U
   902 1177              123 CON    @123          S
   903 1200               72 CON    @72           :
   904 1201             1015 CON    @1015               CR
   905 1202               12 CON    @12                 LF
   906 1203              123 CON    @123          S
   907 1204              111 CON    @111          I
   908 1205              132 CON    @132          Z
   909 1206              105 CON    @105          E
   910 1207               75 CON    @75           =
   911 1210              440 CON    @440                BLANK
   912 1211                1 GOSUB  FNDEND        COMPUTE SIZE
   912 1212                0
   913 1213              116 C=0
   914 1214             1160 DADD=C
   915 1215             1570 C=REGN 13
   916 1216               74 RCR    3
   917 1217             1106 C=A-C  X
   918 1220             1334 PT=    13
   919 1221              320 LC     3
   920 1222                1 GOSUB  PBINBD        PRINT SIZE
   920 1223                0
   921 1224                1 GOSUB  EOLL
   921 1225                0
   922 1226              460 LDI
   923 1227              176 CON    @176
   924 1230                1 GOSUB  CKANGL
   924 1231                0
   925 1232                1 GOSUB  PBYTEC
   925 1233                0
   926 1234                1 GOSUB  PRTMSG        PRINT:LF,SIGMA=
   926 1235                0
   927 1236               75 CON    @75
   928 1237              440 CON    @440
   929 1240             1570 C=REGN 13          COMPUTE SIGMA
   930 1241              674 RCR    11
```

```
1242           246 AC EX   X
1243           574 RCR     6
1244          1106 C=A-C   X
                   LEGAL
1245             1 GOSUB   PBINB0          PRINT SIGMA
1246             0
1247             1 GOSUB   EOLL
1250             0
1251          1670 C=REGN  14              CMP DEG RAD GRAD CODE
1252            74 RCR     3
1253          1434 PT=     1
1254           102 C=0     PT
1255          1530 ST=C
1256          1004 S2=     0
1257             4 S3=     0
1260          1210 S7=     1
1261          1630 C=ST
1262             1 GOSUB   PPROM1          OUTPUT DEG,RAD, OR GRAD
1263             0
1264             1 GOSUB   EOLL
1265             0
1266          1670 C=REGN  14              FIX,SCI,ENG?
1267            74 RCR     3
1270          1530 ST=C
1271           460 LDI
1272           234 CON     @234
1273            14 ?S3=1
1274            57 GOC     OUTDSP (1301)
1275          1046 C=C+1   X
1276          1014 ?S2=1
1277            23 GONC    OUTDSP (1301)
1300          1046 C=C+1   X
1301 OUTDSP    256 AC EX
1302             1 GOSUB   BPROMT          OUTPUT FIX SCI OR ENG
1303             0
1304          1670 C=REGN  14              GET N
1305          1074 RCR     2
1306           132 C=0     M
1307          1074 RCR     2
1310           136 C=0     S
1311          1076 C=C+1   S
                   LEGAL
1312             1 GOSUB   PBINBD          FIX N ETC
1313             0
1314             1 GOSUB   PRTMSG          PRINT:LF,LF,FLAGS:
1315             0
1316          1015 CON     @1015              CR
1317          1015 CON     @1015              CR
1320            12 CON     @12                LF
1321           106 CON     @106            F
1322           114 CON     @114            L
1323           101 CON     @101            A
1324           107 CON     @107            G
1325           123 CON     @123            S
1326           472 CON     @472            :
1327          1670 C=REGN  14              STORE FLAGS AND COUNTER
1330           106 C=0     X
1331 FLGLOP    530 M=C
1332             1 GOSUB   PRTMSG          PRINT LF, F,SPACE
1333             0
```

```
1334          1015 CON    @1015          CR
1335            12 CON    @12            LF
1336           106 CON    @106         F
1337           440 CON    @440           BLANK
1340           630 C=M
1341             1 GOSUB  PBINB0       PRINT NUMBER OF FLAG
1342             0
1343           630 C=M
1344           756 C=C+C               IS FLAG SET
1345           127 GOC    FLGSET (1357) YES
1346             1 GOSUB  PRTMSG       PRINT "  CLEAR"
1347             0
1350           242 CON    @242         TWO BLANKS
1351           103 CON    @103         C
1352           114 CON    @114         L
1353           105 CON    @105         E
1354           101 CON    @101         A
1355           522 CON    @522         R
1356            73 GOTO   LPCHK  (1365)
1357 FLGSET      1 GOSUB  PRTMSG       PRINT "  SET"
1360             0
1361           242 CON    @242         TWO BLANKS
1362           123 CON    @123         S
1363           105 CON    @105         E
1364           524 CON    @524         T
1365 LPCHK    1114 ?S9=1               ANY ERROR ?
1366             1 GSUBC  PECHK
1367             1
1370           630 C=M
1371           246 AC EX  X
1372           460 LDI
1373            14 CON    @14
1374          1546 ? A#C  X
1375            37 GOC    C+C     (1400)
1376          1670 C=REGN 14
1377           674 RCR    11
1400 C+C       756 C=C+C
1401           460 LDI
1402            70 CON    @70          DONE YET
1403           246 AC EX  X            C READY TO STORE IN M
1404          1046 C=C+1  X            INC COUNT
1405          1546 ? A#C  X
1406          1237 GOC    FLGLOP (1331) LOOP AGAIN
                   ENTRY  FINISH
1407 FINISH      1 GOSUB  LPECHK       EOLL, CHECK PRINTER ERRORS
1410             0
1411             1 GOLONG NFRPU
1412             2
*************************************************************
*PRKEYS-PRINTS OUT KEY REASSIGNMENTS
*IF NONE EXIST-PRINTS KEYS: NONE
*OTHERWISE PRINTS 1 1 SIZE
*                 1 5 ASHIFT
*                 4 2 SPCCHS
*************************************************************
1413           223 CON    @223
1414            31 CON    @31
1415             5 CON    @5
1416            13 CON    @13
1417            22 CON    @22
```

```
1420             20 CON    @020
                    ENTRY  PRKEYS
1421 PRKEYS       1 GOSUB  IPRT            INITIALIZE PRINT
1422              0
1423              1 GOSUB  PRTMSL          PRINT "USER KEYS:"
1424              0
1425             15 CON    @15             EOLL (CR )
1426            125 CON    @125            U
1427            123 CON    @123            S
1430            105 CON    @105            E
1431            122 CON    @122            R
1432             40 CON    @40                BLANK
1433            113 CON    @113            K
1434            105 CON    @105            E
1435            131 CON    @131            Y
1436            123 CON    @123            S
1437            472 CON    @472            :
1440            116 C=0
1441           1160 DADD=C
                    ENTRY  KEYLP1
1442 KEYLOP    1150 REGN=C 9               SET INDEX AT 0,0
1443 KEYLP1    1170 C=REGN 9               GET CURRENT INDEX BACK
1444            256 AC EX                  SET UP INDEX FO TBITMP
1445              1 GOSUB  TBITMP          IS THIS KEY ASSIGNED?
1446              0
1447           1356 ? C#0
1450            503 GONC   INCCNT (1520)   NO SKIP PRINTING
1451              1 GOSUB  EOLL            FINISH LAST LINE
1452              0
1453              1 GOSUB  PWAIT           CHECK PRINT ERRORS
1454              0
1455           1170 C=REGN 9
1456            136 C=0    S               SET FOUND ONE BIT
1457           1076 C=C+1  S
1460           1150 REGN=C 9
1461           1474 RCR    1
1462           1530 ST=C
1463             14 ?S3=1                  IS THIS A SHIFTED KEY?
1464              1 GSUBNC PBLANK
1465              0
1466              1 GOSUB  PRKC
1467              0
1470              1 GOSUB  PBLANK
1471              0
1472           1170 C=REGN 9
1473           1474 RCR    1
1474            246 AC EX  X
1475            546 A=A+1  X
1476           1404 S1=    0               GET KEY CODE OR ADR
1477              1 GOSUB  GCPKC
1500              0
1501             14 ?S3=1                  RAM?
1502            127 GOC    DORAM  (1514)   YES
1503             34 PT=    3               XROM FUNCTION
1504           1342 ?C#0   PT
1505             47 GOC    DOXROM (1511)
1506              1 GOSUB  PPROM1          MAINFRAME FCN
1507              0
1510            103 GOTO   INCCNT (1520)
1511 DOXROM       1 GOSUB  PPXROM          XROM FUNCTION
```

```
1512                 0
1513                53 GOTO    INCCNT (1520)

1514 DORAM         416 A=C                           ADDRESS TO A3:0
1515               504 S6=     0                     SAY RAM
1516                 1 GOSUB   PLBL0
1517                 0

                       ENTRY   INCCNT
1520 INCCNT        116 C=0
1521              1160 DADD=C
1522              1434 PT=     1                     ADD 8 TO ROW
1523              1020 LC      8
1524              1434 PT=     1
1525               242 AC EX   PT
1526              1170 C=REGN  9                     GET INDEX BACK
1527              1002 C=A+C   PT                    SHIFTED YET?
1530              1123 GONC    KEYLOP (1442)         DO SHIFTED
1531              1066 C=C+1   XS                    INC COLUMN
1532              1150 REGN=C  9
1533               766 C=C+C   XS
1534               766 C=C+C   XS
1535               133 GONC    KEYLNK (1550)         COL WAS THREE OR LESS
1536               742 C=C+C   PT
1537               742 C=C+C   PT
1540                37 GOC     INCCOL (1543)         YES INC COLUMN
1541              1366 ?C#0    XS
1542                63 GONC    KEYLNK (1550)         COL=4
1543 INCCOL       1170 C=REGN  9                     GET INDEX BACK
1544               126 C=0     XS                    RESET COLUMN
1545              1042 C=C+1   PT                    INC COLUMN
1546              1150 REGN=C  9                     PUT INDEX AWAY
1547               742 C=C+C   PT                    ROW LARGER THAN 7
1550 KEYLNK          1 GOLNC   KEYLP1                NO
1551                 2
1552              1170 C=REGN  9
1553              1376 ?C#0    S                     FIND ANY ASSIGNMENTS
1554                77 GOC     DONKEY (1563)         YES
1555                 1 GOSUB   PRTMSG                NO
1556                 0
1557               116 CON     @116                  N
1560               117 CON     @117                  O
1561               116 CON     @116                  N
1562               505 CON     @505                  E
1563 DONKEY          1 GOLONG  FINISH
1564                 2

*
************************************************************************
************** PRX -- PRINT X REG, NO DISPLAY   ************************
************************************************************************
                       ENTRY   PRX10
                       ENTRY   PRX
1565               230 CON     @230                  X
1566                22 CON     @22                   R
1567                20 CON     @20                   P
1570 PRX             1 GOSUB   IPRT
1571                 0
1572                 1 GOSUB   PRXSUB
1573                 0
```

```
  1140 1574 PRX10      1 GOSUB  PECHK
  1140 1575            0
* CANNOT SIMPLY RETURN HERE BECAUSE 1) PRXSUB CALL USES UP ALL FOUR
* SUBROUTINE LEVELS: NFRPU IS NO LONGER ON THE STACK, AND 2) CARD
* READER ROM LOGIC FOR THE 7PRX FUNCTION DOES A GOSUB TO PRX (VIA
* PRT18) AND DOES NOT WANT PRX TO RETURN TO IT.
  1145 1576            1 GOLONG NFRPU
  1145 1577            2
*
************************************************************************
*-GLINE#= GET LINE #
*
*-CALCULATES LINE # (BINARY) IF THE LINE #= FFF, OTHERWISE RETURNS
* EXISTING LINE #.
*-GENERATES ERROR MESSAGE FOR PRIVATE PROGRAM, & DOESN'T RETURN
*
*-USES:    A, B(0-3), C, M, N, P, Q, (S0-S8),      3 SUB LEVELS
*-INPUTS:  CURRENT PRIVACY FLAG (S12) FOR VALID LINE#,R12=DESIRED PC
*-OUTPUTS: A(X)= C(X)= LINE #  (BINARY)
*-ASSUMES: NOTHING
*
  1159                   ENTRY  GLINE#
  1160 1600 GLINE#     1 GOSUB  LINNUM         GET  LINE #
  1160 1601            0
  1161 1602         1346 ? C#0  X              LINE #= 0?
  1162 1603           27 GOC    GLIN20 (1605) NON-ZERO
  1163 1604         1046 C=C+1  X              YES, INC TO 1
**C= REG 15 ON EXIT FROM LINNUM!!!!!!!!!!!!!!!
  1165 1605 GLIN20 1750 REGN=C 15             STORE NEW LINE #
  1166 1606          406 A=C    X              LINE # TO "A"
  1167 1607         1514 ?S12=1                PRIVATE?
  1168 1610            1 GOLC   ERRPR          YES,ERROR, DISPLAY "PRIVATE"
  1168 1611            3
  1169 1612         1740 RTN
***********************************************************************
*
* OOPMSG - PUT UP "OUT OF PAPER" MESSAGE IN LCD
* USES: C6:0, AND 1 ADDITIONAL SUBROUTINE LEVEL
* IN: NOTHING
* OUT: LEAVES CHIP 0 ENABLED AND SS0 UP
* ASSUMES: NOTHING
*
  1178                   ENTRY  OOPMSG
  1179 1613 OOPMSG     1 GOSUB  MESSLP
  1179 1614            0
  1180 1615           20 CON    @20            P
  1181 1616           22 CON    @22            R
  1182 1617           11 CON    @11            I
  1183 1620           16 CON    @16            N
  1184 1621           24 CON    @24            T
  1185 1622            5 CON    @05            E
  1186 1623           22 CON    @22            R
  1187 1624           40 CON    @40
  1188 1625            5 CON    @05            E
  1189 1626           22 CON    @22            R
  1190 1627           22 CON    @22            R
  1191 1630         1040 CON    @1040
  1192 1631            1 GOSUB  ENCP00
  1192 1632            0
  1193 1633            1 GOSUB  UNL
  1193 1634              0
  1194 1635              1 GOLONG STMSGF
  1194 1636              2
  1195
*******************************************************************
************* ACX -- ACCUMULATE X REG IN PRINTER BUFFER ************
*******************************************************************
  1199                     ENTRY  ACX
  1200 1637            230 CON    @230          X
  1201 1640              3 CON    3             C
  1202 1641              1 CON    1             A
  1203 1642 ACX          1 GOSUB  IACHR
  1203 1643              0
  1204 1644              1 GOSUB  ACXSUB
  1204 1645              0
  1205 1646           1263 GOTO   PRX10  (1574)
*
**********************************************************************
********************  PRT11=  AVIEW  *********************************
**********************************************************************
  1210                     ENTRY  PAVIEW
6F4 1211 1647 PAVIEW     1 GOSUB  CKEN          OK TO PRINT ?
  1211 1650              0
  1212 1651           1740 RTN                  P+1 - NO
  1213 1652              1 GOSUB  FNDPTR        P+2 - YES, SEE IF PTR THERE
  1213 1653              0
  1214 1654             53 GOTO   PAVW10 (1661) NO PRINTER
  1215 1655              1 GOSUB  INITC
  1215 1656              0
  1216 1657              1 GOLONG PRA20
  1216 1660              2
  1217 1661 PAVW10 1304 S13=   0
  1218 1662           1670 C=REGN 14
  1219 1663           1530 ST=C
  1220 1664           1740 RTN
*
*******************************************************************
**+
**+
* CKEN - CHECK PRINTER ENABLED IF RUNNING OR SINGLE-STEPPING
*
* RETURNS TO: P+1 IF NOT OK TO PRINT
*             P+2 IF OK TO PRINT
* USES: C, ST[7:0], S9, PT,   NO ADDITIONAL SUBROUTINE LEVELS
* INPUT: CHIP 0 ENABLED, HEXMODE
* OUTPUT: IF RTN TO P+2 THEN S9=0, CHIP 0 ENABLED, HEXMODE
*
*
*
  1235                     ENTRY  CKEN
  1236 1665 CKEN      1670 C=REGN 14            GET STATUS BITS
  1237 1666           1530 ST=C
  1238 1667           1314 ?S13=1               RUNNING?
  1239 1670             37 GOC    CKEN10 (1673) YES
  1240 1671            114 ?S4=1                SINGLE STEPPING?
  1241 1672             53 GONC   CKEN20 (1677) NOPE
  1242 1673 CKEN10     434 PT=    8
  1243 1674            742 C=C+C  PT
  1244 1675            742 C=C+C  PT            FLAG 21? (PRINTER ENABLED?)
  1245 1676           1640 RTN NC               NO
  1246 1677 CKEN20 1104 S9=     0                CLEAR ERROR FLAG
  1247 1700            1 GOLONG RTNP+2
  1247 1701            2
*
*
  1250                   FILLTO @1701
  1251
*
  1253 1702 KYCKX  1614 ?S0=1                  PRINTER EXIST ?
  1254 1703             63 GONC   KYCKX2 (1711) NO
  1255 1704            144 HPL=CH 1
  1256 1705           1005 CH=    @201           ENABLE FLAG TEST
  1257 1706           1254 SRQR?                 SERVICE REQUEST RECEIVED ?
  1258 1707              1 GOLC   PRSVC          YES, LET'S TAKE A LOOK AT PRINTER
  1258 1710              3
  1259 1711 KYCKX2       1 GOLONG RMCK10
  1259 1712              2
  1260                   FILLTO @1712
* WHEN PAUSING WITH THE PRINTER TURNED OFF, THE EXTRA WORD TIMES TO
* DISCOVER THAT THE PRINTER IS OFF LENGTHEN THE PAUSE BY ABOUT 10%.
  1263                   ENTRY  PRT11
  1264                   ENTRY  PRT6
  1265      PRT18
  1266 1713 CRPRTX       1 GOLONG PRX    6F28   CR: 97 PRTX
  1266 1714              2
  1267      PRT17
  1268 1715 CRPSTK       1 GOLONG PRSTK [illegible]  CR: 97 PRST
  1268 1716              2
  1269      PRT16
  1270 1717 CRPREG       1 GOLONG REGL  6324   CR: 97 PREG
  1270 1720              2
  1271 1721 PRT15        1 GOLONG XPRT15 6C34  SSTBST
  1271 1722              2
  1272 1723 PRT14        1 GOLONG ENDALP 639E  ENTERING OR EXITING ALPHA MODE
  1272 1724              2
  1273 1725 PRT13        1 GOLONG OVERFL [illegible]  D.E. UNDERFLOW OR OVERFLOW
  1273 1726              2
  1274 1727 PRT12        1 GOLONG PRTCAT 6345  PRINT CATALOG IN TRACE
  1274 1730              2
5FD9 1275 1731 PRT11        1 GOLONG PAVIEW 6F47
  1275 1732              2
5FDB 1276 1733 PRT10        1 GOLONG PVIEW  6435
  1276 1734              2
5FDD 1277 1735 PRT9         1 GOLONG PADV   6C4D
  1277 1736              2
  1278 1737 PRT8         1 GOLONG DATA&R 658E  DATA ENTRY STRING & R/S
  1278 1740              2
  1279 1741 PRT7         1 GOLONG PPROMP 62A6
  1279 1742              2
  1280 1743 PRT6         1 GOLONG PMESSG 6E4C  PRINT MESSAGES
  1280 1744              2
  1281 1745 PRT5         1 GOLONG DATA&F 6592  DATA ENTRY STRING & FUNCTION
  1281 1746              2
  1282                                         (NUT040  OR  NAME42)
  1283 1747 PRT4         1 GOLONG DATAPR 6421  KEY SEQUENCE ABORTED
  1283 1750              2
  1284                                         OR PAUSE EXPIRED
  1285                                         OR RAK100 IN CN1
  1286 1751 PRT3         1 GOLONG ALPHOP 6BE5  BEGIN TO KEY IN ALPHA OPERAND
  1286 1752              2
```

```
      1287 1753 PRT2        1 GOLONG NXINST 652B    NEXT INST TO BE XEQ,RUNNING PGM
      1287 1754             2
      1288 1755 PRT1        1 GOLONG PXTR   6C0E
      1288 1756             2
  *
      1290                    FILLTO @1757
           1757          0000 NOP
      1291                    ENTRY  ACRGCX
  6FF01292 1760 ACRGCX      1 GOLONG ACREGC 64CE    SEND C REG TO PRINTER
      1292 1761             2
      1293                    ENTRY  PBYTCX
      1294 1762 PBYTCX      1 GOLONG PBYTEC 6528    SEND C1:0 TO PRINTER
      1294 1763             2
      1295 1764 PPAUSE   1163 GOTO   KYCKX  (1702)  ENTRY FROM PAUSE LOOP
      1296 1765 PRUN        0 NOP                   RUNNING
      1297 1766 WAKEP       0 NOP                   WAKE UP FROM DEEP SLEEP W/O KEY
      1298 1767 POWOFP      0 NOP
      1299 1770 I/OSVP   1123 GOTO   KYCKX  (1702)
      1300 1771 DEEPSP      0 NOP                   WAKE-UP FROM DEEP SLEEP
      1301 1772 COLDSP      0 NOP                   COLD START ENTRY POINT
      1302 1773 PRTID       5 CON    @05            E
      1303 1774            62 CON    @62            2
      1304 1775            14 CON    @14            L
      1305 1776            20 CON    @20            P
      1306 1777 CKSUMP      0 NOP                   PRINTER CHECKSUM
      1307                    END

      ERRORS :        0
```

SYMBOL TABLE

```
ACCHR     135   -
ACCHRX    137   -
ACCOL     161   -
ACRGCX   1760   -
ACSPCC    555   -    552
ACSPEC    536   -
ACX      1642   -
ADV01     402   -    376
ADV02     413   -    422    420    366
ADV03     423   -
ADV04     424   -    405
ADV10     454   -    465
ADV20     466   -    462
ADV30     470   -    455
ADV50     473   -    423    416
ADVCKC    377   -    373
ADVKEY    367   -
AERRDE    541   -
BLD10     515   -    513
BLDSPC    506   -
C+C      1400   -   1375
CKEN     1665   -
CKEN10   1673   -   1670
CKEN20   1677   -   1672
CKSUMP   1777   -
CKTRC1    211   -    202
CKTRCE    174   -
COLDSP   1772   -
CPBYT1   1042   -   1036
CPBYTE   1030   -   1026
CRPREG   1717   -
CRPRTX   1713   -
CRPSTK   1715   -
DEEPSP   1771   -
DONKEY   1563   -   1554
DORAM    1514   -   1502
DOXROM   1511   -   1505
EOLREX    126   -     63
FILLIN      1   -
FILLNP      0   -
FINISH   1407   -
FLGLOP   1331   -   1406
FLGSET   1357   -   1345
GLIN20   1605   -   1603
GLINE#   1600   -
I/OSVP   1770   -
IACHR     646   -
IACOL     660   -
IAUALL    663   -
IAUNA     662   -
IAUNB     666   -
IN20      647   -    661
IN40      674   -    667
IN999     644   -    651    637
INADV     655   -
INADXP     12   -
```

```
INCCNT   1520    -   1513 1510 1450
INCCOL   1543    -   1540
INIT10    715    -    706  657
INIT12    731    -    634
INIT15    732    -    720
INIT20    737    -    733
INIT30    741    -    736
INIT35    746    -    742
INIT40    750    -    745
INIT5     700    -
INIT50    755    -    751
INIT60    757    -    754  747  744  740  735  731
INIT70    767    -    765
INIT80    772    -    770
INITC     702    -    677  643
INITSC    630    -
INITSM    631    -
IPRT      635    -
KEYLNK   1550    -   1542 1535
KEYLOP   1442    -   1530
KEYLP1   1443    -
KYCKX    1702    -   1770 1764
KYCKX2   1711    -   1703
LPCHK    1365    -   1356
NOPTR     603    -    622
OOPMSG   1613    -
OUTDSP   1301    -   1277 1274
PAD      1110    -
PAD1+A   1107    -
PADV      115    -
PAVIEW   1647    -
PAVW10   1661    -   1654
PBYA+C   1112    -
PBYT01   1040    -   1105
PBYT05   1067    -   1063 1055
PBYT06   1070    -   1066
PBYT10   1074    -   1101
PBYT11   1102    -   1106
PBYT12   1104    -   1075
PBYTCS   1047    -    775  773
PBYTCX   1762    -
PBYTDU   1045    -
PBYTEC   1050    -   1113 1044
PE05      617    -    602
PE10      621    -    654  642  575
PE30      623    -    577
PECHK     570    -
PECHKJ    172    -    122  107
PEDIAG    573    -
PKEY      275    -    246
PKEY15    301    -    277
PKEY35    334    -    330
PMESSG   1114    -
PMSG10   1121    -   1117
PMSG15   1140    -   1133
PMSG16   1142    -   1130
PMSG20   1147    -   1145
POWCFP   1767    -
PPAUSE   1764    -
PPECHK    167    -    153  151
```

```
PRBUF      102   -
PRFLAG    1164   -
PRKC       776   -
PRKC10    1006   -     777
PRKC20    1027   -    1024
PRKEYS    1421   -
PRSVC      213   -
PRT1      1755   -
PRT10     1733   -
PRT11     1731   -
PRT12     1727   -
PRT13     1725   -
PRT14     1723   -
PRT15     1721   -
PRT16     1717   -
PRT17     1715   -
PRT18     1713   -
PRT2      1753   -
PRT3      1751   -
PRT30      316   -     311
PRT4      1747   -
PRT40      342   -     353
PRT5      1745   -
PRT50      354   -     351
PRT6      1743   -
PRT60      362   -     343
PRT7      1741   -
PRT8      1737   -
PRT9      1735   -
PRTID     1773   -
PRUN      1765   -
PRX       1570   -
PRX10     1574   -    1646
PRXSUB      52   -
PSVC10     236   -     231
PSVC20     241   -     235
PSVC30     245   -     233
PSVC80     252   -     225
PSVC90     261   -     256     240     227     221
PSVC95     263   -
PSVC99     272   -     270
PXTR        17   -
PXTR2       31   -      26
PXTR4       37   -      30
PXTREX      42   -      36      32
SPEC-K     721   -
SPEC10     553   -     565
TMRMSG    1134   -
UNLEX      625   -     673     671
WAKEP     1766   -
XPRT15      64   -
```

```
ENTRY TABLE

ACCHR     135    -
ACCHRX    137    -
ACCOL     161    -
ACRGCX   1760    -
ACSPCC    555    -
ACSPEC    536    -
ACX      1642    -
ADV50     473    -
ADVKEY    367    -
BLDSPC    506    -
CKEN     1665    -
CKTRCE    174    -
CPBYTE   1030    -
FILLIN      1    -
FILLNP      0    -
FINISH   1407    -
GLINE#   1600    -
IACHR     646    -
IACOL     660    -
IAUALL    663    -
IAUNA     662    -
IAUNB     666    -
INADV     655    -
INADXP     12    -
INCCNT   1520    -
INIT5     700    -
INIT60    757    -
INITC     702    -
INITSC    630    -
INITSM    631    -
IPRT      635    -
KEYLP1   1443    -
OOPMSG   1613    -
PAD      1110    -
PAD1+A   1107    -
PADV      115    -
PAVIEW   1647    -
PBYA+C   1112    -
PBYTCX   1762    -
PBYTDU   1045    -
PBYTEC   1050    -
PECHK     570    -
PEDIAG    573    -
PMESSG   1114    -
PRBUF     102    -
PRFLAG   1164    -
PRKC      776    -
PRKC20   1027    -
PRKEYS   1421    -
PRSVC     213    -
PRT11    1731    -
PRT50     354    -
PRT6     1743    -
PRX      1570    -
PRX10    1574    -
PRXSUB     52    -
```

```
PXTR      17   -
SPEC-K   721   -
XPRT15    64   -
```

```
 FNDPTR     22    105    120    223    573    640    652    663   1126   1652
 FNDPTR     23    106    121    224    574    641    653    664   1127   1653
 FNSTS      37    345    413    457
 FNSTS      40    346    414    460
 GCPKC    1477
 GCPKC    1500
 IACHR     140   1642
 IACHR     141   1643
 IACOL     164    543
 IACOL     165    544
 IAUNB    1131
 IAUNB    1132
 INADV     111    124
 INADV     112    125
 INCADA     13
 INCADA     14
 INIT60    710
 INIT60    711
 INITC      50   1655
 INITC      51   1656
 INITSM    400
 INITSM    401
 IPRT     1164   1421   1570
 IPRT     1165   1422   1571
 KEYLP1   1550
 KEYLP1   1551
 LDSST0     33    175
 LDSST0     34    176
 LEFTJ     316    437
 LEFTJ     317    440
 LINNUM   1600
 LINNUM   1601
 LPECHK    113   1407
 LPECHK    114   1410
 MESSL     303    313    432
 MESSL     304    314    433
 MESSLP   1613
 MESSLP   1614
 MSG110   1153
 MSG110   1154
 MSGA      363
 MSGA      364
 MSGNL     365
 NFRPU    1411   1576
 NFRPU    1412   1577
 NLT040    360
 NLT040    361
 NXINST   1753
 NXINST   1754
 OOPCHK    655    702
 OOPCHK    656    703
 OOPMSG    236    600
 OOPMSG    237    601
 OVERFL   1725
 OVERFL   1726
 PAD         6
 PAD         7
 PADV     1735
 PADV     1736
 PAVIEW   1731
```

```
PAVIEW  1732
PBINBC  1245   1341
PBINBC  1246   1342
PBINBD  1222   1312
PBINBD  1223   1313
PBLANK  1464   1470
PBLANK  1465   1471
PBYTDU   562
PBYTDU   563
PBYTEC   170    471    723    727   1232   1762
PBYTEC   171    472    724    730   1233   1763
PECHK    172   1366   1574
PECHK    173   1367   1575
PILERR   623
PILERR   624
PLBL0   1516
PLBL0   1517
PLEREX   603
PLEREX   604
PMESSG  1743
PMESSG  1744
PPROMI  1262   1506
PPROMI  1263   1507
PPROMP  1741
PPROMP  1742
PPXROM  1511
PPXROM  1512
PR15RT    73
PR15RT    74
PRA20   1657
PRA20   1660
PRBUF    241
PRBUF    242
PRKC    1466
PRKC    1467
PRKC20  1013
PRKC20  1014
PRSTK   1715
PRSTK   1716
PRSTKX    46
PRSTKX    47
PRSVC   1707
PRSVC   1710
PRT50    466
PRT50    467
PRTCAT  1727
PRTCAT  1730
PRTLCD  1134
PRTLCD  1135
PRTMSG    55    424   1234   1314   1332   1346   1357   1555
PRTMSG    56    425   1235   1315   1333   1347   1360   1556
PRTMSL  1166   1423
PRTMSL  1167   1424
PRX     1713
PRX     1714
PRXSUP  1572
PRXSUP  1573
PVIEW   1733
PVIEW   1734
PWAIT   1453
PWAIT   1454
PXTR    1755
PXTR    1756
RDFMER  1102
RDFMER  1103
REGL    1717
REGL    1720
RMCK10   273  1711
RMCK10   274  1712
RFECHK   126
RFECHK   127
RTNP+2   211  1700
RTNP+2   212  1701
SPEC-K   374
SPEC-K   375
STMSGF  1635
STMSGF  1636
TBITMP  1445
TBITMP  1446
UNL       42   263   625  1140  1633
UNL       43   264   626  1141  1634
UNLRSF   354   473   571
UNLRSF   355   474   572
XPRT15  1721
XPRT15  1722

End of VASM assembly
```